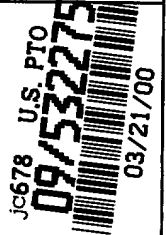


<h1 style="text-align: center;">UTILITY PATENT APPLICATION TRANSMITTAL</h1> <p><i>(Only for new nonprovisional applications under 37 CFR 1.53(b))</i></p>	Attorney Docket No. 826.1595/JDH
	First Named Inventor or Application Identifier:  Masaki UKAI
	Express Mail Label No.

<b>APPLICATION ELEMENTS</b> See MPEP chapter 600 concerning utility patent application contents.	ADDRESS TO: <b>Assistant Commissioner for Patents</b> <b>Box Patent Application</b> <b>Washington, DC 20231</b>
---	---

1. ☒ Fee Transmittal Form
2. ☒ Specification, Claims & Abstract ..... [ Total Pages: 63 ]
3. ☒ Drawing(s) (35 USC 113) ..... [ Total Sheets: 32 ]
4. ☒ Oath or Declaration ..... [ Total Pages: 3 ]
  - a. ☒ Newly executed (original or copy)
  - b. ☐ Copy from a prior application (37 CFR 1.63(d)) (for continuation/divisional with Box 17 completed)
    - i. ☐ **DELETION OF INVENTOR(S)**  
Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
5. ☐ Incorporation by Reference (usable if Box 4b is checked)  
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
6. ☐ Microfiche Computer Program (Appendix)
7. ☐ Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
  - a. ☐ Computer Readable Copy
  - b. ☐ Paper Copy (identical to computer copy)
  - c. ☐ Statement verifying identity of above copies



### ACCOMPANYING APPLICATION PARTS

8. ☒ Assignment Papers (cover sheet & document(s))
9. ☐ 37 CFR 3.73(b) Statement (when there is an assignee) [ ] Power of Attorney
10. ☐ English Translation Document (if applicable)
11. ☒ Information Disclosure Statement (IDS)/PTO-1449 [X] Copies of IDS Citations
12. ☐ Preliminary Amendment
13. ☒ Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
14. ☐ Small Entity Statement(s) [ ] Statement filed in prior application, status still proper and desired.
15. ☒ Certified Copy of Priority Document(s) (if foreign priority is claimed)
16. ☐ Other:

### 17. If a CONTINUING APPLICATION, check appropriate box and supply the requisite information:

[ ] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application No: \_\_\_\_ / \_\_\_\_

### 18. CORRESPONDENCE ADDRESS

STAAS & HALSEY, LLP  
 Attn: James D. Halsey, Jr.  
 700 Eleventh Street, N.W., Suite 500  
 Washington, DC 20001

Telephone: (202) 434-1500  
 Facsimile: (202) 434-1501

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Application of: )  
)  
Masaki UKAI )  
) Group Art Unit: To Be Assigned  
Serial No.: To Be Assigned )  
)  
Filed: March 20, 2000 ) Examiner: To Be Assigned  
)  
For: DEVICE AND METHOD FOR )  
CONTROLLING WRITING OF )  
BRANCH HISTORY INFORMATION )

**PRELIMINARY AMENDMENT**

Assistant Commissioner of Patents and Trademarks  
Washington, D.C. 20231

Sir:

Before examination of the above-identified application, please amend the application as follows:

**IN THE CLAIMS:**

**Claim 8,** lines 1 and 2, change "any one of claims 4 through 7" to --claim 1--.

**Claim 15,** line 1, change "claim 1 or 10" to --claim 1--.

**Claim 39,** line 1, change "claim 26 or 34" to --claim 26--.

**REMARKS**

This Preliminary Amendment is submitted to improve the form of the specification and claims as originally filed.


Variable	Unit	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100																																																																																																																																																																																						
Population	Millions	12.1	12.2	12.3	12.4	12.5	12.6	12.7	12.8	12.9	13.0	13.1	13.2	13.3	13.4	13.5	13.6	13.7	13.8	13.9	14.0	14.1	14.2	14.3	14.4	14.5	14.6	14.7	14.8	14.9	15.0	15.1	15.2	15.3	15.4	15.5	15.6	15.7	15.8	15.9	16.0	16.1	16.2	16.3	16.4	16.5	16.6	16.7	16.8	16.9	17.0	17.1	17.2	17.3	17.4	17.5	17.6	17.7	17.8	17.9	18.0	18.1	18.2	18.3	18.4	18.5	18.6	18.7	18.8	18.9	19.0	19.1	19.2	19.3	19.4	19.5	19.6	19.7	19.8	19.9	20.0	20.1	20.2	20.3	20.4	20.5	20.6	20.7	20.8	20.9	21.0	21.1	21.2	21.3	21.4	21.5	21.6	21.7	21.8	21.9	22.0	22.1	22.2	22.3	22.4	22.5	22.6	22.7	22.8	22.9	23.0	23.1	23.2	23.3	23.4	23.5	23.6	23.7	23.8	23.9	24.0	24.1	24.2	24.3	24.4	24.5	24.6	24.7	24.8	24.9	25.0	25.1	25.2	25.3	25.4	25.5	25.6	25.7	25.8	25.9	26.0	26.1	26.2	26.3	26.4	26.5	26.6	26.7	26.8	26.9	27.0	27.1	27.2	27.3	27.4	27.5	27.6	27.7	27.8	27.9	28.0	28.1	28.2	28.3	28.4	28.5	28.6	28.7	28.8	28.9	29.0	29.1	29.2	29.3	29.4	29.5	29.6	29.7	29.8	29.9	30.0	30.1	30.2	30.3	30.4	30.5	30.6	30.7	30.8	30.9	31.0	31.1	31.2	31.3	31.4	31.5	31.6	31.7	31.8	31.9	32.0	32.1	32.2	32.3	32.4	32.5	32.6	32.7	32.8	32.9	33.0	33.1	33.2	33.3	33.4	33.5	33.6	33.7	33.8	33.9	34.0	34.1	34.2	34.3	34.4	34.5	34.6	34.7	34.8	34.9	35.0	35.1	35.2	35.3	35.4	35.5	35.6	35.7	35.8	35.9	36.0	36.1	36.2	36.3	36.4	36.5	36.6	36.7	36.8	36.9	37.0	37.1	37.2	37.3	37.4	37.5	37.6	37.7	37.8	37.9	38.0	38.1	38.2	38.3	38.4	38.5	38.6	38.7	38.8	38.9	39.0	39.1	39.2	39.3	39.4	39.5	39.6	39.7	39.8	39.9	40.0	40.1	40.2	40.3	40.4	40.5	40.6	40.7	40.8	40.9	41.0	41.1	41.2	41.3

It is respectfully requested that this Preliminary Amendment be entered in the above-references application.

If any fees are required in connection with the filing of this Preliminary Amendment, please charge same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS &amp; HALSEY

  
James D. Halsey  
Registration No.

James/D. Halsey, Jr.  
Registration No.: 22,729

700 Eleventh Street, N.W.  
Washington, D.C. 20001  
(202) 434-1500

Date: March 20, 2000

APPLICATION FOR  
UNITED STATES LETTERS PATENT  
SPECIFICATION

Inventor(s): Masaki UKAI

Title of the Invention: DEVICE AND METHOD FOR  
CONTROLLING WRITING OF BRANCH  
HISTORY INFORMATION

007250" 5/22/85

## DEVICE AND METHOD FOR CONTROLLING WRITING OF BRANCH HISTORY INFORMATION

### Background of the Invention

5

### Field of the Invention

The present invention relates to a device for  
controlling the writing of branch history information  
in an information processing apparatus provided with  
10 a branch estimation unit.

### Description of the Related Art

In an instruction execution processing apparatus,  
performance is improved by using technologies  
15 including a pipeline process and sequentially starting  
the execution of subsequent instructions without  
waiting for the completion of a specific instruction.  
In this case, if a previous instruction is an  
instruction to change the execution sequence of a  
20 subsequent execution, such as a branch instruction,  
etc., and a branch is established, the execution  
pipeline is impeded, and in the worst case, the  
performance is degraded if an instruction on the  
branch destination is not inputted to the execution  
25 pipeline. Under these circumstances, a branch

prediction unit represented by a branch history is provided and the establishment of a branch is predicted. If a branch establishment is predicted, the performance could be improved by inputting an instruction on the branch destination following a branch instruction to an execution control unit or instruction process unit.

However, in a conventional branch prediction unit, the branch history information of a branch instruction of which the execution is completed in a branch control unit was registered in a branch history by temporarily stopping the instruction fetch.

According to this system, in particular, if a branch prediction fails and a correct subsequent instruction is re-executed, and specifically, if a re-instruction fetch occurs, an instruction fetch pipeline is temporarily stopped by the branch history information writing the completed branch instruction into the branch history although the frequency of a fetch request is high since a temporary instruction buffer is empty. Accordingly, the performance was not improved.

Specifically, although, as in the execution of a branch instruction BC shown in Fig. 1, in the cycle W of the execution cycle of a branch instruction, the

branch history information is written into the branch history, only one access can be allowed at one time since the branch history is comprised of RAMs (random access memory). Accordingly, the re-instruction fetch  
5 of the branch destination instruction is started one clock cycle behind the cycle W of the branch instruction.

The branch history information is to be written in the branch history when it can be completed in the  
10 branch control unit, and is not written at the time of actual branch execution completion. Accordingly, if an interruption occurs immediately before the branch execution, a return address stack is sometimes operated by mistake.

15 If there is a short loop, especially in an instruction string as shown in Fig. 2A, this instruction string can be fetched at one instruction fetch. Therefore, the timing becomes as shown in Fig. 2B, and the branch history information is written  
20 later than the re-instruction fetch. If there is a branch instruction to be looped immediately after the re-instruction fetch, etc., a correct branch cannot be predicted. In this case, since one more re-instruction fetch is executed, the performance is  
25 degraded.

### Summary of the Invention

An objective of the present invention is to provide an instruction execution control device for suppressing a process delay in an information processing apparatus provided with a branch prediction unit, and a method thereof.

A device in the first aspect of the present invention comprises a control unit controlling in such a way that the writing of branch history information into a branch prediction unit and the control over the memory unit may not occur simultaneously in a branch history information write control device in an instruction execution processing apparatus provided with a memory unit storing an instruction string, etc., and a branch prediction unit performing the branch prediction of a branch instruction.

In a branch history information write control device of an instruction execution processing apparatus provided with a branch prediction unit performing the branch prediction of a branch instruction, a device in the second aspect of the present invention comprises a return address stack unit and a control unit controlling in such a way that if the branch instruction is not executed although the branch instruction is an instruction corresponding to



a sub-routine call or return, and the write request of the branch history information of the branch instruction is issued to the branch prediction unit, the branch history information is written in the branch prediction unit, but the return address stack unit may not be operated.

A method in the first aspect of the present invention is an instruction control method in an apparatus provided with a memory storing an instruction string, etc., and a branch prediction unit performing the branch prediction of a branch instruction, and comprises the step of controlling in such a way that the writing of branch history information in the branch prediction unit and the control over the memory may not occur simultaneously.

A method in the second aspect of the present invention is an instruction control method in an apparatus provided with a branch prediction unit performing the branch prediction of a branch instruction and a return address stack, and comprises the step of controlling in such a way that if the branch instruction is not executed although an instruction corresponding to the sub-routine call or return obtained as an execution result of the branch instruction issues the write request on the branch

history information to the branch prediction unit, the branch history information is written in the branch prediction unit, but the return address stack is not operated.

5           Although conventionally an instruction fetch is held and branch history information is written in the branch prediction unit (branch history), according to the present invention, an instruction fetch is not held and the branch history is written in a branch  
10 history in a timing such that the writing of the branch history information in the branch history and the control of the instruction fetch, etc., over a memory may not occur simultaneously. Accordingly, no instruction fetch is held and thereby the execution  
15 process speed can be improved.

#### **Brief Descriptions of the Drawings**

Fig. 1 shows conventional problems occurring at the time of execution of a branch instruction.

20           Figs. 2A and 2B show conventional problems occurring when a short loop is formed between branch instructions.

Fig. 3 shows effects obtained when a branch instruction is executed according to the present  
25 invention.

Figs. 4A and 4B show effects obtained when a short loop is formed between branch instructions according to the present invention.

Fig. 5 shows the basic configuration of the preferred embodiments of the present invention.

Fig. 6 shows an example configuration of a branch history.

Fig. 7 shows an example basic configuration for delaying the writing of the branch history information of a branch instruction re-instruction-fetched.

Fig. 8 shows an example detailed configuration of a block 31 shown in Fig. 7.

Fig. 9 shows an example detailed configuration of a block 32 shown in Fig. 7.

Fig. 10 shows an example of a circuit corresponding to a block 30 shown in Fig. 7.

Fig. 11 is a timing diagram of a preferred embodiment using a counter.

Fig. 12 shows another preferred embodiment of a circuit corresponding to a box 30 shown in Fig. 7.

Fig. 13 shows an example circuit configuration for operating the device even if a temporary instruction buffer is empty.

Fig. 14 shows another preferred embodiment of a circuit of a box 30 shown in Fig. 7 (No. 1).

Fig. 15 shows another preferred embodiment of a circuit of a box 30 shown in Fig. 7 (No. 2).

Fig. 16 shows an example configuration of a branch history write reservation station.

5 Fig. 17 shows the overall operation flow of the control over writing in a branch history when a reservation station is used.

10 Fig. 18 shows a selection circuit for plurally and simultaneously writing in a branch history (No. 1).

Fig. 19 shows a selection circuit for plurally and simultaneously writing in a branch history (No. 2).

15 Fig. 20 shows a selection circuit for plurally and simultaneously writing in a branch history (No. 3).

Fig. 21 shows a selection circuit for plurally and simultaneously writing in a branch history (No. 4).

20 Fig. 22 shows a circuit for plurally and simultaneously writing in a branch history.

Fig. 23 shows an example circuit configuration of a circuit for compulsorily writing in a branch history.

25 Fig. 24 shows an example of the valid circuit of

a write reservation station.

Fig. 25 shows an example configuration of a reservation station in the case where writing in a reservation station is available after the completion of an instruction execution under IID (Instruction ID) control.

Fig. 26 shows an example configuration of the CSE valid circuit of a write reservation station.

Fig. 27 shows the data storage configuration of a write reservation station.

Fig. 28 shows an example configuration of a CSE valid circuit in the case where a branch prediction unit is provided with a return address stack.

Fig. 29 shows an example circuit for nullifying the entry of a reservation station if an instruction execution is temporarily stopped due to an interruption, etc.

Fig. 30 shows one preferred embodiment of a bypass hit circuit.

#### Description of the Preferred Embodiments

In the preferred embodiment of the present invention, the writing of branch history information is held while a branch history is being looked up, and the branch history information is written in a cycle

which does not look up the branch history. If this method is used, any impediment of an instruction fetch disappears. Accordingly, a pipeline process can be smoothly performed and thereby the degradation of the performance described earlier can be prevented.

Specifically, as shown in Fig. 3, if an instruction string shown in Fig. 1 is executed, in this preferred embodiment, an instruction fetch of NOP5, which was conventionally started following the cycle W of a branch instruction BC, can be started in the same cycle as the cycle W of the branch instruction BC. Accordingly, by adopting this preferred embodiment, the performance can be improved by one clock cycle on average.

While the density of an instruction fetch is high immediately after a re-instruction fetch, writing into a branch history is held. In particular, since a new branch instruction is executed and completed for several clock cycles immediately after a re-instruction fetch, most of performance degradation can be prevented in a small-scale circuit without providing a reservation station described next.

The preferred embodiment of the present invention provides a method for realizing a temporary buffer in order to hold the writing of branch history

information. If instruction fetch requests continue, and, even if an instruction pipeline process is very smoothly executed, the impediment of an instruction fetch request can be eliminated by providing a reservation station as a holding device for realizing such a buffer function. Accordingly, the performance degradation can be prevented.

If there is a new write request while branch history information is being held by the holding device, a control method for preventing a reservation station which becomes a temporary buffer, from overflowing is provided. According to such a preferred embodiment, all branch history information can be registered/updated in a branch history without failure.

Depending on the configuration of a branch history, a plurality of branch histories can be simultaneously written in the same cycle. In a branch history comprised of a plurality of RAMs (random access memory) which can store another entry, a maximum of the same number of different entries as RAMs composing the branch history can be written. It is clear that by doing so, a lot of history information of branch instructions can be written in a branch history while the impediment of writing for

branch prediction is reduced.

Branch history information is reported before a branch instruction is completed. Generally speaking, in an instruction execution control device adopting a system, such as an out-of-order, etc., the branch instruction is sometimes held because the previous instruction is not completed yet although all conditions for executing a specific instruction are met. In particular, in this case, after the uncompleted instruction meets all conditions, instructions which have been held are collectively attempted to be completed. Accordingly, by adopting such a method, the load of writing in the branch history can be dispersed.

If in the case of a branch prediction unit provided with a return address stack, the return address stack is operated when the branch history information of a branch instruction which has not actually been executed is reported, the correspondences between pairs of subsequent sub-routine calls and returns naturally collapse, and the prediction of the branch destination address of a sub-routine return instruction fails. Therefore, the return address stack is configured not to be operated by the call/return instruction mistakenly being



executed as long as the return address stack is actually executed, although the branch instruction is registered in a branch history.

The registration/update of the branch history information is held until an instruction execution is completed. In this case, the branch history information can be written simultaneously with the execution completion or the writing can be held until there is a favorable timing. In this case, alternatively, a method for identifying an instruction with an ID number assigned for each instruction can be used to judge whether the execution of the branch instruction is completed. If the execution of the branch instruction is cancelled although the branch history information is transmitted, the relevant entry corresponding to the write reservation station can also be nullified. In this case, since the history information of a branch that has been executed with certainty is to be used if the writing is held until the execution is completed, no incorrect branch history information is registered and thereby the accuracy of a branch prediction is improved.

If a branch prediction is performed by a branch prediction unit, branch history information, writing of which is held, etc., can also be used by bypassing

the writing. In this case, since bypassing was performed, the latest branch history information can be used, and an enhanced effect can be obtained, in particular in a local loop. Specifically, the process  
5 delay of a pipeline can be improved by approximately several clock cycles for each short loop.

For example, if, as shown in Figs. 4A and 4B, short loops shown in Figs. 2A and 2B are executed, conventionally, as shown in Figs. 2A and 2B, the  
10 instruction fetch of the instruction L(3) of the third loop is started in a cycle B of the execution cycle of the instruction L(2) of the second loop. However, in Figs. 4A and 4B, the instruction fetch of the instruction L(3) is started in the timing of the cycle  
15 IB of the instruction L(2) by bypassing the writing and using the branch history information. In this way, according to this preferred embodiment, performance can be improved by six clock cycles compared with the conventional method shown in Figs. 2A and 2B.

20 As the RAMs of a branch history, dual port RAMs which can perform writing and reading independently in the same cycle, are used. If there is room in circuit mounting, the performance degradation can be prevented most easily using this method.

25 Fig. 5 shows the basic configuration of the

preferred embodiments of the present invention.

The instruction execution control device of the present invention adopts an out-of-order system. Therefore, in Fig. 5, which concerns an execution control unit, only approximate time dependency is shown.

In this preferred embodiment, it is assumed that an instruction fetch can secure an instruction string of 16 bytes which is 8-byte-aligned, at one request.

First, the address of an instruction to be fetched next is inputted to a selector 11. The addresses to be inputted to the selector 11 are the sequential instruction address which is inputted from an adder 10, the start address for an interruption process, the prediction branch destination instruction address which is the branch prediction result inputted from a branch history, and the branch destination instruction address which is confirmed by a branch instruction process unit 17 executing a branch instruction. The selector 11 is controlled by an instruction execution flow control device (not shown in Fig. 5) which is controlled by an instruction completion process unit 19 for guaranteeing the order of instructions in an out-of-order type information processing apparatus, and is configured to select an

appropriate address in each case. The address  
outputted from the selector 11 is inputted to a branch  
history 20 as the effective address IF\_EAG of the  
instruction fetch, and simultaneously is inputted to  
5 an instruction fetch address register 12 to perform  
the instruction fetch. The instruction fetch address  
register 12 inputs an instruction corresponding to the  
address given by the selector 11, and simultaneously  
the address of the instruction is inputted to the  
10 adder 10. After being stored in a temporary  
instruction buffer 14, the instruction inputted to an  
instruction cache 13 is decoded in a decoder 15 and  
transmitted to the branch instruction process unit 17,  
an execution instruction process unit 18, and other  
15 instruction process units to be processed. In the case  
of a branch instruction, the execution of branch  
conditions, etc., is performed in an address  
calculating unit 16, and the result is inputted to the  
branch instruction process unit 17. If the branch  
20 instruction of the execution is not determined and the  
execution is described by means of an instruction  
string, the execution is performed by the execution  
instruction process unit 18, and the execution result  
is inputted to the branch instruction process unit 17.  
25 Furthermore, the execution completion reports of the

branch instruction process unit 17 and execution instruction process unit 18 are processed by an instruction completion process unit 19, and are used to guarantee an instruction execution order and to control an out-of-order type instruction execution order. The address of a branch destination instruction which is determined as an execution result of the branch instruction is inputted to the selector 11 from the branch instruction process unit 17.

Fig. 6 shows an example configuration of a branch history.

The branch history 20 is comprised of two RAMs 20-1 and 20-2, and these two RAMs 20-1 and 20-2 are configured to search for branch history information in the ranges of high-order 8 bytes and low-order 8 bytes, respectively, using an 8-byte-aligned instruction fetch address. According to this preferred embodiment, the branch history 20 is provided with a return address stack to predict the return destination of a sub-routine return instruction, which is not shown in Fig. 6.

According to this preferred embodiment, the branch instruction process unit 17 in the execution process unit can handle a maximum of four branch instructions, a maximum of one out of four branch

instructions is simultaneously prepared for the execution completion, and the branch instruction is transmitted to the instruction completion control unit 19. According to the prior art, branch history information was transmitted to the branch history in this timing.

The instruction completion control unit 19 assigns IIDs to all instructions being executed (from instruction decoding until completion), and manages the execution. In this preferred embodiment, it is assumed that a maximum of 16 instructions can exist in the execution unit. Specifically, it is assumed that any value between 0 and 15 is assigned as the value of the IID.

It is preferable to use as the branch history dual port RAMs which can perform reading and writing simultaneously.

Fig. 7 shows an example basic configuration for delaying the writing of the branch history information of a branch instruction re-instruction-fetched. Figs. 8 and 9 show example detailed configurations of the blocks 31 and 32 shown in Fig. 7.

According to this preferred embodiment, if it is found in the branch instruction process unit 17 that a branch prediction has failed, the branch instruction

At this time, if an instruction cache 13 cannot accept the request for some reason, the re-instruction fetch request is held until the request can be accepted. While the re-instruction fetch request is being held, +REIFCH\_REQUEST remains ON. In this case, usually this signal is directly inputted to the block 30. However, a circuit shown in Fig. 13, which is described later, is provided and a signal outputted from this circuit can also be inputted to the block 30 instead.

25           If the re-instruction fetch request  
          (+REIFCH REQUEST of "H") is inputted to the block 30,

both a signal obtained by logically inverting a signal for holding the writing of branch history information in the branch history 20 (-WAIT\_BR\_COMP\_WRITE) and a signal for holding the re-instruction fetch signal of a re-instruction fetch (+BR\_COMP\_REIFCH\_HOLD) are outputted. +BR\_COMP\_REIFCH\_HOLD is inputted to the block 31. -WAIT\_BR\_COMP\_WRITE is inputted to the block 32. The block 31 receives branch information, such as +BR\_COMP\_AS\_TAKEN, etc., and further, a branch instruction address (+BR\_COMP\_IAR) and a branch destination instruction address (+BR\_COMP\_TIAR) are inputted to the block 31. Then, a signal for controlling an instruction to write in the branch history (+CREATE\_NEW\_ENTRY, +UPDATE\_OLD\_ENTRY, +ERASE\_ENTRY), branch history information, a signal obtained by latching a branch destination instruction address (+BR\_COMP\_IAR\_LCH), and a signal obtained by latching a branch destination instruction address (+BR\_COMP\_TIAR\_LCH) are outputted from the block 31. The branch history information and branch destination instruction address are inputted to the port for writing data of the branch history 20. The branch instruction address is inputted to the selector 34 together with an instruction fetch address (+IF\_EAG). A signal for instructing to write in the branch



history 20 is outputted from the block 32. By this  
 signal, an appropriate address is outputted from the  
 selector 34. Simultaneously, the branch history 20 is  
 made to enter a writable state, and the branch history  
 5 information, etc., can be written.

Fig. 8 shows an example circuit detail of the  
 block 31 shown in Fig. 7, which stores data while  
 writing is being held up.

A signal obtained by logically inverting a signal  
 10 indicating whether the update of the entry of the  
 branch history 20 is valid (-BRHIS\_UPDATE\_VALID) and  
 +BR\_COMP\_REIFCH\_HOLD are inputted to an AND circuit  
 41. Therefore, if the update of the entry of a branch  
 history is invalid and a re-instruction fetch is  
 15 requested, branch history information, a branch  
 instruction address and a branch destination  
 instruction address are stored in a latch circuit 42.  
 The branch history information, branch instruction  
 address and branch destination instruction address are  
 20 outputted from the latch circuit 42. As described with  
 reference to Fig. 7, simultaneously,  
 +CREATE\_NEW\_ENTRY, +UPDATE\_OLD\_ENTRY, +ERACE\_ENTRY,  
 which are signals for instructing to write in the  
 branch history 20, are extracted from the branch  
 25 history information and are outputted from the latch

Fig. 9 shows the circuit detail of the block 32 shown in Fig. 7, which generates a signal for controlling branch history writing.

Fig. 10 shows an example of a circuit corresponding to a block 30 shown in Fig. 7. Fig. 11 is a timing diagram showing the operation of the circuit shown in Fig. 10.

(see Fig. 11(d)), writes branch history information in a branch history (see Fig. 11(e)) and returns to an execution waiting (re-instruction fetch request waiting) state. Since by using this method, priority is given to these three instruction fetch requests over a normal re-instruction fetch, this method is applicable to a case where the temporary instruction buffer 14 is empty as well as to a temporary instruction fetch request. In this case, an output shown in Fig. 13 can be inputted instead of a signal +REIFCH\_REQUEST, which is one of the input signals shown in Figs. 7 and 10.

Specifically, the counter shown in Fig. 10 is comprised of two bits, and if +REIFCH\_REQUEST is inputted, a signal of positive logic and a signal of inverted logic are inputted to one terminal of an AND circuit 61 and one terminal of an OR circuit 62, respectively, via a latch circuit 60. In Fig. 10, <0> is a high-order bit and <1> is a low bit. If a re-instruction fetch signal is "H", the value stored in a latch circuit 63 becomes "01", and is maintained. If the re-instruction fetch signal becomes "0", "0" and "H" are outputted from the terminal Q of the latch circuit 60 and the inverted terminal of Q, respectively. Therefore, if a counter value "01" is

outputted via an EXOR circuit 64, an inverter 65, an OR circuit 66 and AND circuits 67 and 68, both of the counter values outputted from the AND circuit 61 and OR circuit 62 become "10". Since "10" is stored in the latch circuit 63, the output of the EXOR circuit becomes "1", and the outputs of the inverter 65 and OR circuit 66 also become "1". Therefore, both of the outputs of the AND circuits become "1". Then, the re-instruction fetch signal becomes "0". Therefore, "11" and "10" are inputted to the AND circuit 61 and OR circuit 62, respectively, and the value of the counter of two bits becomes "11". If in this way, the re-instruction fetch signal becomes "0", the counter starts counting up to "11". When the count value reaches "11", the count value returns to "00". In response to this count value, the EXOR circuit 69 outputs a re-instruction fetch hold signal +BR\_COMP\_REIFCH\_HOLD when the count value is "01" or "10" (see Fig. 11D). When the count value is "00" or "11", a signal for permitting to write in the branch history 20 -WAIT\_BR\_COMP\_WRITE is outputted.

As described above, if a re-instruction fetch signal of "H" is inputted, the count value changes from "0" to "1", and if the re-instruction fetch signal drops to "L", the count value is counted up.

Then, if the count value becomes "3" two clock cycles later, the re-instruction fetch hold signal (+BR\_COMP\_REIFCH\_HOLD) and the signal for permitting to write in the branch history 20 (+WRITE\_BRHIS\_VALID) become "L" and "H", respectively, and thereby writing in the branch history becomes available.

Fig. 12 shows another preferred embodiment of a circuit corresponding to the box 30 shown in Fig. 7.

In Fig. 12, if a valid instruction is decoded in an instruction decoder 15, a signal +D\_VALID becomes ON. The re-instruction fetch signal (+REIFCH\_REQUEST) and a signal +D\_VALID are provided to a set/reset flip-flop 81 as a set signal and a reset signal, respectively. In this way, the output signal of this flip-flop 81 is maintained from when a re-instruction fetch request is issued until an instruction string corresponding to the request is decoded. Therefore, while the output signal of the flip-flop is ON, the signal +BR\_COMP\_REIFCH\_HOLD is ON, and -WAIT\_BR\_COMP\_WRITE (+BR\_COMP\_WRITE), which is a signal obtained by inverting a signal outputted from an inverter 82 becomes OFF.

Fig. 13 shows an example circuit configuration for operating the device even if a temporary instruction buffer is empty.

A circuit shown in Fig. 13 provides a signal which replaces +REIFCH\_REQUEST, as the input of the box 30 shown in Fig. 7. Specifically, (1) if an instruction buffer 14 is empty (+I\_BUFF\_EMPTY is "H"),  
 5 (2) if in a cycle IT, an instruction fetch request is invalid (-IT\_IF\_REQ\_VALID is "H") and (3) if in a cycle IB, the instruction fetch request is invalid (-IB\_IF\_REQ\_VALID is "H"), the output of an AND circuit 91 becomes "H". Therefore, if the three conditions,  
 10 (1) to (3), above are met or if there is a re-instruction fetch request (+REFECH\_REQUEST is "H"), a signal +REIFCH\_REQ\_OR\_IF\_EMPTY becomes ON. Then, this signal is inputted to the box 30 shown in Fig. 7 instead of a re-instruction fetch signal.

15 Figs. 14 and 15 show another preferred embodiment of the box 30 shown in Fig. 7.

Fig. 14 shows an example configuration of the box 30 in the case where if there is a re-instruction fetch, branch history information is not written in the branch history 20, but if the instruction cache  
 20 13 cannot accept an instruction fetch, the branch history information is written in the branch history 20. In this case, not an instruction fetch request, but a signal +SU\_BUSY, which is transmitted from the  
 25 instruction cache 13, is inputted as input. The fact

that this signal is "H" means that since the instruction cache 13 is full, and an instruction fetch cannot be accepted. In this case, if a signal +SU\_BUSY of "H" is inputted, a write permit signal of "H" (+BR\_COMP\_WRITE) is outputted from a buffer 101 to the branch history 20. If a signal +SU\_BUSY of "H" is not inputted, a signal for instructing to temporarily hold a re-instruction fetch of "L" (+BR\_COMP\_REIFCH\_HOLD) is outputted from an inverter 102.

Fig. 15 shows an example circuit configuration of the box 30 shown in Fig. 7 in the case where if there is a request on a real instruction fetch, such as the pre-fetch of an instruction, branch history information is written in the branch history 20. In this case, if a signal, such as an instruction pre-fetch request, etc., is inputted as input, and this signal is "H", a signal for permitting to write in a branch history of "H" (+BR\_COMP\_WRITE) is outputted from a buffer 111. If the signal is "L", a signal for instructing to temporarily hold a re-instruction fetch of "H" (+BR\_COMP\_REIFCH\_HOLD) is outputted.

Fig. 16 shows an example configuration of a branch history write reservation station.

When an entry is registered in a reservation station, a valid flag (Valid) is set, and by using the

signal as a hold signal, branch prediction data (branch history information) is held.

If a reservation station 120 (120-1) shown in Fig. 16 is used, writing in the branch history 20 can also be controlled by using one of the preferred embodiments described above.

The reservation station 120 is comprised of four entries (RSW0-RSW3). In each entry RSWx (x=0-3), a valid flag (Valid) and an instruction address (IAR) are registered, and further branch history information 121 corresponding to the registered instruction is stored.

Fig. 17 shows the overall circuit configuration of a preferred embodiment which controls writing in the branch history 20 in the case where the reservation station 120 shown in Fig. 16 is used.

First, branch history information (+BR\_COMP\_AS\_TAKEN, etc.) is inputted to the reservation station 120 by the branch instruction execution unit, and simultaneously both a branch instruction address (+BR\_COMP\_IAR) and a branch destination instruction address (+BR\_COMP\_TIAR) are also inputted. The reservation station 120 is configured as shown in Figs. 16 and 22-25. From the reservation station 120, data to be written in a



branch history for odd addresses 20-1 and a branch history for even addresses 20-2 are outputted via a selector 130 shown in Fig. 21 (+BRW\_ODD/EVEN\_DATA). Signals outputted from the reservation station 120 are

5 processed in a block 140 to generate an odd address branch history write permit signal (+RSW\_ODD\_WRITE\_VAL) and an even address branch history write permit signal (+RSW\_ODD\_WRITE\_VAL), respectively, which become signals for permitting to

10 write in the branch histories 20-1 and 20-2, respectively. In a selector 150, either an address IAR transmitted from the reservation station 120 or an address for retrieving from the branch history 20 (+IF\_EAG) is selected. In particular, if a write permit signal is received from the block 140, the

15 selector 150 selects the IAR and makes a certain circuit write in the branch history 20. For example, the writing in the branch history 20 is performed by a circuit shown in Fig. 22.

20 In this preferred embodiment, plurally and simultaneous writing is also performed.

Figs. 16-20 show control circuits for plurally and simultaneously writing in the branch history 20.

In this preferred embodiment, the branch history

25 is comprised of two RAMs. Therefore, if there are two



bytes and if there are 16 or more different write addresses, the addresses are written in the low-order 8 bytes and the high-order 8 bytes of the branch histories 20-1 and 20-2, respectively. Then, the device is controlled in such a way that the combinations of two addresses with priority are written in the branch history 20 with priority in a priority determination circuit 143. If the signal of one of the AND circuits 142-1 - 142-6 is ON, a plurally writable signal (+WRITE\_DOUBLE) is outputted.

A circuit 140B located in the lower section of Fig. 18 is a write signal selection circuit used when one entry of the reservation station 120 is written into the branch history 20. If either of the valid signals of the entry of each reservation station RSWx is ON, a signal indicating that data to be written is available (+WR\_DATA\_AVAILABLE) is outputted. Priority is set to the writing of each entry of the reservation station 120 by a circuit 146. If a plurally writable signal is OFF and +WR\_DATA\_AVAILABLE is "H", a single write signal (+WRITE\_SINGLE) is outputted and simultaneously a signal indicating which entry of the reservation station 120 should be written (+WRITE\_RSWx) is outputted.

In Fig. 18, the output of this AND circuit can

also be used instead of the signal +RSWx\_VALID by providing an AND circuit 149 as shown in the upper left section of Fig. 18. In this case, the signal +RSWx\_VALID and a signal -RSW\_CSE\_VALID, which is described later, are inputted. After an instruction execution is completed, writing in the branch history 20 can be performed using these signals. Specifically, if the instruction is not completed, the signal +RSW\_CSE\_VALID is ON. How to generate this signal is described later.

Fig. 19 shows a selection circuit for plurally and simultaneously writing in a branch history (No. 2).

In Fig. 19, the output from the single write selection circuit 140B shown in Fig. 18 and the 28th bit of an instruction address (IAR) outputted from the reservation station 120 are inputted to AND circuits 151 (151-1 - 151-4). By checking whether a write signal inputted from the single write selection circuit and the 28th bit of an instruction address are ON or OFF, a write permit signal used when the instruction address is odd, is outputted. Then, if one of the AND circuits 151-1 - 151-4 is ON, the writing of an odd address is permitted. Therefore, if this signal is outputted, a signal for permitting to write

in the odd branch history 20-1 (+RSW\_ODD\_WRITE\_VALID) is outputted. If the double write signal (+WRITE\_DOUBLE) is ON, writing in the odd branch history 20-1 and even branch history 20-2 are permitted. Specifically, +RSW\_ODD\_WRITE\_VALID and +RSW\_EVEN\_WRITE\_VALID are turned ON. If the writing in the odd branch history 20-1 is OFF, a signal to be inputted to an AND circuit 66 becomes ON. Furthermore, if the single write signal (+WRITE\_SINGLE) is ON, a signal for permitting to write in the even branch history 20-2 (+RSW\_EVEN\_WRITE\_VALID) becomes ON. If either a signal indicating that writing in the branch history 20 is not held (-WAIT\_BR\_COMP\_WRITE) or a signal for compulsorily writing in the branch history 20 (+FORCE\_WRITE\_BRHIS) is ON, this signal for permitting to write in the odd or even branch history is outputted. +FORCE\_WRITE\_BRHIS is described later.

Figs. 20 and 21 show selection circuits for plurally and simultaneously writing in branch histories (No. 3) (No. 4).

A circuit shown in Fig. 20 generates select signals for writing in the entries RSW0-RSW3 of the No. 1 through No. 3 branch histories of the reservation station 120 (+SEL\_RSWx\_WRITE) using the output shown in Fig. 18 as input, and inputs the

signals to a circuit shown in Fig. 21.

The circuit shown in Fig. 21 selects data transmitted from the reservation station 120 (RSW<sub>x</sub>\_DATA). In Fig. 21, a selection circuit for an odd branch history 20-1 and a selection circuit for an even branch history 20-2 are separately configured. Data from the reservation station 120 are inputted to multiplexers 181 and 182, and one of the pieces of data is transmitted as RSW\_OLD\_DATA or RSW\_EVEN\_DATA.

The multiplexer 181 adds a select signal (+SEL\_RSW<sub>x</sub>\_WRITE), which is the output of the circuit shown in Fig. 20, to the 28th bit of an instruction address (IAR) inputted from the RSW0 to RSW3 of the reservation station 120, and switches the operation based on the result. In this way, if the select signal is ON and the instruction address is odd, a select signal is transmitted and corresponding data are transmitted.

In the same way, the multiplexer 182 adds a select signal, which is the output of the circuit shown in Fig. 20, to a signal obtained by logically inverting the 28th bit of an instruction address inputted from the RSW0 to RSW3 of the reservation station 120, and switches the operation based on the result. In this way, if the select signal is ON and

the instruction address is even, corresponding data are transmitted.

Fig. 22 shows a circuit for plurally and simultaneously writing in a branch history.

5        An address for writing an instruction in the odd branch history 20-1 (RSW\_ODD\_IAR) and an address for reading from the branch history (IF\_EAG) are inputted to a selector 191. In the same way, an address for writing an instruction in the even branch history 20-2  
10        (RSW\_EVEN\_IAR) and an address for reading from the branch history 20 (IF\_EAG) are inputted to a selector 192. +RSW\_ODD\_WRITE\_VALID, which is the output of the circuit shown in Fig. 19, is also inputted to the selector 191. The device is configured in such a way  
15        that if writing in the odd branch history 20-1 is permitted, the selector 191 selects RSW\_ODD\_IAR. Similarly, the device is configured in such a way that if +RSW\_EVEN\_WRITE\_VALID, which is the output of the circuit shown in Fig. 19, is inputted and writing in  
20        the even branch history is permitted, the selector 192 selects the RSW\_EVEN\_IAR. Write-enable is set in the odd branch history 20-1 and even branch history 20-2 by +RSW\_ODD\_WRITE\_VALID and +RSW\_EVEN\_WRITE\_VALID, respectively, and RSW\_ODD\_DATA and RSW\_EVEN\_DATA,  
25        respectively, are written.

Furthermore, if in this preferred embodiment, the reservation station 120 is full and data to be written are further transferred from the branch instruction control unit, at least one piece of data being stored  
5 can be written. Figs. 23 and 24 show this preferred embodiment.

Fig. 23 shows an example circuit configuration of a circuit for compulsorily writing in a branch history.

10 If the valid signal of an entry from a circuit shown in Fig. 24 (+RSWx\_VALID) is inputted to an AND circuit 201 and all the entries of all the reservation stations 120 are valid, the reservation stations are full. Therefore, in this case, a signal indicating  
15 this +RSW\_FULL is outputted. If the update of the branch history 20 is valid, +BRHIS\_UPDATE\_VALID inputted from the branch instruction process unit 17 becomes ON, a signal for compulsorily writing in a reservation station +FORCE\_WRITE\_BRHIS is generated  
20 and used in the operation shown in Fig. 19.

Fig. 24 shows an example configuration of the valid circuit of a reservation station 120.

If the update of the branch history 20 is valid (+BRHIS\_UPDATE\_VALID is ON) and the entry RSW0 of the  
25 No. 1 branch histories of a reservation station 120



is invalid (-RSW0\_VALID is ON) or if the reservation station is full (+RSW\_FULL) and the No. 0 branch history of the reservation station 120 is selected (+SEL\_RSW0\_WRITE, which is the output shown in Fig. 20, is ON), a set signal is transmitted to a flip-flop 211-1, and a signal indicating that the No. 0 branch history of the reservation station 120 is valid (+RSW\_VALID) becomes ON. In the same way, the same setting is made in the No. 1 through No. 3 branch histories of the reservation station 120. However, priority is set according to the ascending order of the number of the reservation station 120 by a priority setting circuit 212. If writing in the branch history 20 is not held (-WAIT\_BR\_COMP\_WRITE is ON) and writing in the reservation station 120 is selected next, the flip-flops 211-1 - 211-4 reset the valid signal of the entry of the reservation station 120. The output of the circuit shown in Fig. 24 becomes the input to the circuit shown in Fig. 23.

Specifically, a signal for holding writing by +WAIT\_BR\_COMP\_WRITE shown in Fig. 19 is suppressed by a signal +FORCE\_WRITE\_BRHIS, and writing in the branch history 20 is compulsorily performed. Simultaneously, the entry of data to be written is registered. By adopting the interleave write system described above,

a maximum of two piece of data can be written in this preferred embodiment.

For example, according to a set associative system, a lot of writing can also be simultaneously performed by controlling a way by +CREATIVE\_NEW, 5 +UPDATE\_OLD\_ENTRY, +ERACE\_ENTRY or another way selecting signal, etc.

Fig. 25 shows an example configuration of a reservation station in the case where writing in the reservation station is available after the completion of an instruction execution under IID (Instruction ID) control. 10

In Fig. 25, in addition to the configuration shown in Fig. 16, the reservation station 120 (120-2) is configured to register both a CSE-Valid flag, which 15 is described later, and the IID of the instruction completion process unit 19. By doing this, a signal RSWx\_CSE\_VAL shown in Fig. 26 can be utilized, and whether an instruction is executed and completed can be judged. 20

A CSE-Valid flag is turned off by comparing the ID of an executed and completed instruction inputted from the instruction complete process unit 19 (COMMIT\_BR\_IID) with IID entries in the reservation station 120. If an instruction execution is 25

temporarily stopped due to an interruption, etc., the CSE-Valid flag is compulsorily turned off by a signal +FLUSH\_RS. By doing this, an entry corresponding to an instruction for which the execution could not be completed due to an interruption, etc., does not remain.

If the CSE-Valid flag is ON, the execution of an instruction is not completed yet. Therefore, a signal obtained by suppressing a signal +RSWx\_VALID using this flag can be transmitted to a write selection circuit instead of +RSWx\_VALID shown in Fig. 18.

Fig. 26 shows an example configuration of the CSE valid circuit of a write reservation station.

The update valid signals of the branch history 20 of the respective entries RDW0-RSW3 of the No. 1 through No. 3 branch histories of the reservation station 120 (+BRHIS\_UPDATE\_VALID) are ON and the reservation station 120 is invalid (-RSWx\_VALID is ON) or if the reservation stations are full (+RSW\_FULL is ON) and the reservation station 120 is selected (+SEL\_RSWx\_WRITE), concerning the No. 0 branch history of the reservation station 120, a signal is inputted to the set ports of flip-flops 221-1 - 221-4, and a signal +RSWx\_CSE\_VALID is outputted. The same process applies to the No. 1 through No. 3 of the reservation

station 120. However, in these cases, a priority circuit 223 is provided in mid-course. This priority circuit 223 is configured in such a way that priority decreases in ascending order of the number of reservation stations.

If COMMIT\_BR\_IID inputted from the instruction completion process management unit 19 in a cycle W and RSWx\_IID recorded in the reservation station 120 are compared by comparators 222 (222-0 - 222-3) and if the signals match, the flip-flops 221-1 - 221-4 are reset. If a signal +FLUSH\_RS outputted at the time of re-instruction fetch is inputted, the flip-flops 221-1 - 221-4 are also reset. Alternatively, if -RSWx\_VALID, which is a signal obtained by logically inverting the output of the circuit shown in Fig. 24, is ON, the flip-flops 221-1 - 221-4 are reset.

If in a circuit shown in Fig. 26, a signal -RSWx\_VALID is simultaneously inputted to OR circuits 224-0 - 224-3 and AND circuits 225-0 - 225-3, there is a possibility that a set signal and a reset signal may be simultaneously inputted. In this case, the circuit is configured in such a way that priority is given to the set signal.

Fig. 27 shows the data storage configuration of the write reservation station shown in Fig. 25.

5           Here, x corresponds to the digits 0 through 3,  
and the same number of the circuit shown in Fig. 27  
as the entries of the reservation station 120,  
specifically, the number of entries of No. 1 through  
No. 3 branch histories of the reservation station 120  
10 are provided.

15       A return address stack is configured to operate  
if a completed branch instruction (to write history  
information) corresponds to a sub-routine call  
instruction or sub-routine return instruction.

When there is a temporary execution stoppage due to an interruption, etc. (a signal +RS1 is ON), it is sufficient if information indicating that the instruction corresponds to the sub-routine call or return of the branch history information which exists in the reservation station 120 is nullified instead of turning a Valid signal (the CSE-Valid flag of the reservation station 120) (see Fig. 25) off. It is

sufficient if the branch history information is written in a timing available at this point. Therefore, the CSE-Valid flag is reset.

+SET\_RSWx\_CSE\_VAL and +RST\_RSWx\_CSE\_VAL shown in  
 5 Fig. 28 correspond to signals inputted to the respective set and reset terminals of the set/reset flip-flops 221-1 - 221-4 shown in Fig. 26.

Two portions including the flip-flops 241-1 and 241-2 located in the lower section are extracted to  
 10 indicate that the portions are the sub-routine call instruction and sub-routine return instruction of the branch history information shown in Fig. 27.

Specifically, if +SET\_RSWx\_CSE\_VAL is inputted, the set/reset flip-flop 242 is set and +RSWx\_CSE\_VAL  
 15 is turned ON. If +RST\_RSWx\_CSE\_VAL is inputted or +RS1 becomes ON, +RSWx\_CSE\_VAL is reset and turned OFF.

If +RS1 is ON and +RSWx\_CSE\_VAL is also ON, the output of an AND circuit 244 becomes "0" and the flip-flops 241-1 and 241-2 are reset. If +RSWx\_VALID, which  
 20 is the output of the circuit shown in Fig. 24, is OFF and the instruction is a sub-routine call instruction, +BR\_COMP\_SUBROUTINE\_CALL becomes ON and is inputted to the flip-flops 241-1 and 241-2. In this case, if the instruction is a sub-routine return instruction,  
 25 +BR\_COMP\_SUBROUTINE\_RTN becomes ON and is inputted to

the flip-flops 241-1 and 241-2.

Specifically, if an instruction execution is not completed (+RSWx\_CSE\_VAL = 0) or there is no temporary execution stoppage due to an interruption, etc., (+RS1 = 0) and the entry of the reservation station 120 is valid (+RSWx\_VALID = 1), an ON signal is inputted to the terminal IH of the flip-flops 241-1 and 241-2, and a signal indicating that the instruction is a sub-routine call instruction or a sub-routine return instruction is stored and outputted as +RSWx\_SUBROUTINE\_CALL or +RSWx\_SUBROUTINE\_RTN.

Fig. 29 shows an example circuit for nullifying the entry of the reservation station if an instruction execution is temporarily stopped due to an interruption, etc.

+SET\_RSWx\_VAL, which is a signal to be inputted to the set terminal of each of the flip-flops 211-1 - 211-4 shown in Fig. 24, is inputted to the set terminal of a set/reset flip-flop 251 and is outputted as +RSWx\_VALID. If +RST\_RSWx\_VAL to be inputted as the reset signal shown in Fig. 24 is inputted or an execution is temporarily stopped due to an interruption, etc., (+RS1 = 1) and an instruction execution is not completed yet (+RSWx\_CSE\_VAL = 1), the set/reset flip-flop 251 is reset.

Fig. 30 shows one preferred embodiment of a bypass hit circuit.

An address comparison (branch prediction) unit 263 comprised of seven comparators 265 (265-1 - 265-7) varies depending on the nature of the branch prediction method of the branch history 20 and should be appropriately configured by a person having ordinary skill in the art. Therefore, only a brief description is made of the address comparison unit 263. Writing is bypassed and data in the write reservation station 120 and branch prediction unit (RSBRx in Fig. 30) can be designated as a search target by generating a branch destination address BRHIS\_TIAR.

Specifically, the address comparison unit 263 compares an inputted address IF\_EAG requested to be read with the branch history 20 (BRHIS-RAMs #0 and #1), the entries #RSW0-3 of the reservation station 120 and an instruction addresses IAR which are outputted from the branch prediction unit #RSBRx. If these respective compared signals match, the address comparison unit 263 outputs corresponding branch destination address TIAR from selection circuits 267-1 - 267-7. Then, one of these signals is selected by a selection circuit 268 and is outputted as the branch



destination address of the branch history (BRHIS\_TIAR). If in the branch prediction unit, a branch destination address generated inside is valid and the branch prediction succeeds, the address  
5 comparison unit outputs a branch destination instruction address to the selection circuit 267-7.

By combining the configurations described above, the performance of the information processing apparatus can be improved.

10 Although in the preferred embodiments described above, a lowercase x is attached to the ends of symbols, like RSWx for example, in the case of a reservation station, this means that a lowercase x can take digits 0 through 3 and there are configurations  
15 and signals corresponding to the respective digits. In other cases, an x suffix means that there are the same number of configurations and signals corresponding to symbols with the suffix as the number indicated by the suffix, and represents the  
20 configurations and signals.

According to the present invention, in the execution of a branch instruction, an instruction fetch request can be processed with priority by delaying the timing of writing of an entry in a branch  
25 history, and the process delay can thereby be avoided.

Furthermore, according to the present invention, if the branch history of a branch instruction which has not been actually executed is reported, the branch history is registered, but a return address stack is not operated. Therefore, the correspondence between a subsequent sub-routine call and a subsequent sub-routine return is prevented from collapsing, and thereby failure in the prediction of the branch destination address of a sub-routine return instruction can be reduced.

00120"522E50



writing in said branch prediction unit the branch history information about a branch instruction which has failed in a branch prediction, said control unit writes the branch history information in said branch prediction unit after several clock cycles (several states).

5. The device according to claim 1, wherein when writing in said branch prediction unit the branch history information about a branch instruction which has failed in a branch prediction, said control unit writes the branch history information in said branch prediction unit after a re-instruction fetch request by the branch instruction is executed and several clock cycles (several states) after the re-instruction fetch request is executed.

6. The device according to claim 1, wherein if the instruction execution processing apparatus is provided with a temporary instruction buffer unit temporarily storing an instruction string outputted from said memory unit,

said control unit writes the branch history information of the branch instruction in said branch prediction unit several clock cycles (several states)

after there is a write request of a branch instruction if the temporary instruction buffer unit is empty and there is no instruction fetch request.

5 7. The device according to claim 1, wherein if the instruction execution processing apparatus is provided with a temporary instruction buffer unit temporarily storing an instruction string outputted from said memory unit,

10 said control unit does not promptly write a branch history of a branch instruction to be requested to be written in said branch prediction unit, waits for a next instruction fetch request and writes the branch history information of the branch instruction  
15 several clock cycles (several states) after the instruction fetch request is executed if the temporary instruction buffer unit is empty and there is not even one instruction fetch request.

20 8. The device according to any one of claims 4 through 7, wherein said control unit uses a counter to count the several clock cycles (several states).

25 9. The device according to claim 1, wherein when writing the branch history information of the branch



write reservation station unit is full and there is a request for registering in the write reservation station unit, said control unit writes in said branch prediction unit at least one group of branch history information, writing of which in the write reservation station unit is held and the branch history information of which has been requested to be registered.

10 14. The device according to claim 10, wherein if said  
branch prediction unit is configured to simultaneously  
write a plurality of entries and said write  
reservation station unit stores a plurality of valid  
information, writing of which is held, said control  
15 unit simultaneously writes the plurality of  
information in a timing such that writing in said  
branch prediction unit is possible.

15. The device according to claim 1 or 10, wherein if  
20 an instruction is conditionally encoded or branched  
by an execution completion of an execution  
instruction, etc., which exits before a branch  
instruction, there is another branch instruction  
before the branch instruction when a branch  
25 destination address is confirmed, and even if the

branch instruction cannot be completed, said control unit writes the branch history information of the branch instruction in said branch prediction unit or registers the information in the write reservation station unit.

16. The device according to claim 15, wherein said control unit provides a flag indicating that the branch history information is written or registered in the write reservation station unit for each corresponding branch instruction being processed.

17. A branch history information write control device in an instruction execution processing apparatus provided with a branch prediction unit performing a branch prediction of a branch instruction, comprising:

a return address stack unit; and

a control unit writing branch history information in the branch prediction unit, but controlling in such a way not to operate the return address stack if the branch instruction corresponds to a sub-routine call or return and the branch instruction is not executed although a write request of the branch history information of the branch instruction is issued to the branch prediction unit.



18. The device according to claim 10, wherein said control unit writes the branch history information, writing of which in the write reservation station unit is held when an execution of an instruction is completed.

19. The device according to claim 10, wherein said control unit writes branch history information of a corresponding entry in said branch prediction unit or said write reservation station unit when an execution of an instruction is completed.

20. The device according to claim 10, wherein if the instruction execution processing apparatus is provided with a unit controlling an execution completion of an instruction in its instruction control unit,

said control unit stores an ID assigned for each instruction, which is stored in the execution completion management unit, in an entry of the write reservation station unit.

21. The device according to claim 10, wherein if it is confirmed that a branch instruction corresponding to a valid entry of the write reservation station unit is neither executed nor completed due to an occurrence

of interruption, etc., the entry corresponding to the write reservation station unit is nullified.

22. The device according to claim 1, further comprising:

a bypass unit making branch history information, writing of which in said branch prediction unit is held, a research target of a branch prediction.

23. The device according to claim 10, further comprising:

a bypass unit making branch history information of a branch instruction which is being executed in its branch execution unit including the write reservation station unit, a research target of a branch prediction.

24. The device according to claim 23, wherein said bypass unit makes the branch history information a search target of a branch prediction when a conditional code for the branch instruction is confirmed if it is confirmed that the branch instruction is not branched and when a branch destination address is confirmed if it is confirmed that the branch instruction is branched.

25. The device according to claim 1, wherein a dual-port RAM in which writing and reading can be simultaneously executed independently is used for said branch prediction unit to hold an entry.

5

26. An instruction control method in an apparatus provided with both a memory storing an instruction string, etc., and a branch prediction unit performing a branch prediction of a branch instruction, comprising:

10

controlling in such a way that writing of branch history information in said branch prediction unit and control over the memory do not occur simultaneously.

15

27. The method according to claim 26, wherein the branch history information is written in said branch prediction unit in a timing such that said memory cannot accept an instruction fetch request.

20

28. The method according to claim 26, wherein the branch history information is written in said branch prediction unit in a timing of requesting a pre-fetch of an instruction.

25

29. The method according to claim 26, wherein if the



to be written is written in said branch prediction unit several clock cycles (several states) after a write request is issued.

- 5 32. The method according to claim 26, further comprising:

temporary instruction buffer step temporarily storing an instruction string, etc., outputted by said memory,

- 10 wherein if there is no instruction string to be stored in the temporary instruction buffer step and there is no instruction fetch request, the branch history information of a branch instruction to be requested to be written is not promptly written in said branch  
15 prediction unit, waits for a next instruction fetch request and is written several clock cycles (several states) after the instruction fetch request is executed.

- 20 33. The method according to claim 26, wherein when the branch history information of a branch instruction which has failed in a branch prediction is written in said branch prediction unit, the branch history information is written after its instruction decoding  
25 unit or said temporary instruction buffer receives a



information, writing of which is held in the write reservation station step and the branch history information which has been requested to be registered, is written.

5

38. The method according to claim 34, wherein if said branch prediction unit is configured to simultaneously write a plurality of entries and the write reservation station unit stores a plurality of valid information, writing of which is held, a plurality of writing executions are performed in a timing such that writing in said branch prediction unit is available.

39. The method according to claim 26 or 34, wherein if an instruction is conditionally encoded or branched by an execution completion of an execution instruction, etc., which exists before a branch instruction, there is another branch instruction before the branch instruction when a branch destination address is confirmed and even if the branch instruction cannot be completed, the branch history information of the branch instruction is written in said branch prediction unit or registered in the reservation station step.

25

40. The method according to claim 39, wherein a flag indicating that the branch history information is written or registered in said write reservation station step is provided for each corresponding branch instruction being processed.

41. An instruction control method in an apparatus provided with both a branch prediction unit performing a branch prediction of a branch instruction and a return address stack, comprising:

controlling in such a way to write branch history information in said branch prediction unit, but not to operate the return address stack if a branch instruction has not been executed although a request for writing of the branch history information in the branch prediction unit is issued by an instruction corresponding to a sub-routine call or return obtained as an execution result of the branch instruction.

42. The method according to claim 34, wherein branch history information, writing of which is held in said write reservation station step, is written when an execution of an instruction is completed.

43. The method according to claim 34, wherein branch



history information of a corresponding entry is written in said write reservation station unit when an execution of an instruction is completed.

5 44. The method according to claim 34, wherein an instruction control unit further comprises the step of managing an execution completion of an instruction, and stores an ID assigned for each instruction which is stored in said execution completion management  
10 step, in an entry in said write reservation station step.

45. The method according to claim 34, wherein if it is found that a branch instruction corresponding to  
15 a valid entry stored in said write reservation station step due to an occurrence of interruption, etc., is not executed and completed, a corresponding entry stored in said write reservation station step is nullified.

20 46. The method according to claim 26, wherein branch history information, writing of which in said branch prediction unit is held, is a search target of a branch prediction.

25

47. The method according to claim 34, wherein branch history information of a branch instruction which is being executed in said write reservation station step, is a search target of a branch prediction.

5

48. The method according to claim 47, wherein the branch history information is a search target of a branch prediction when a conditional code for the branch instruction is confirmed if it is confirmed that the branch instruction is not branched, and when a branch destination address is confirmed if it is confirmed that the branch instruction is branched.

10

**Abstract of the Disclosure**

5 If a re-instruction fetch request is inputted to  
a box 30, a signal for holding writing in a branch  
history or a signal for holding a re-instruction fetch  
is generated. A signal for holding the re-instruction  
fetch, branch history information, a branch  
instruction address and a branch destination  
instruction address are inputted to the box 31. The  
10 box 31 outputs these data, and simultaneously  
generates a new entry in the branch history, updates  
an old entry or generates a signal for instructing to  
erase an entry. These data are inputted to the box 32,  
and a signal for instructing whether to write a branch  
15 instruction address, a branch destination instruction  
address or branch history information are generated.

001220" 5/22/65

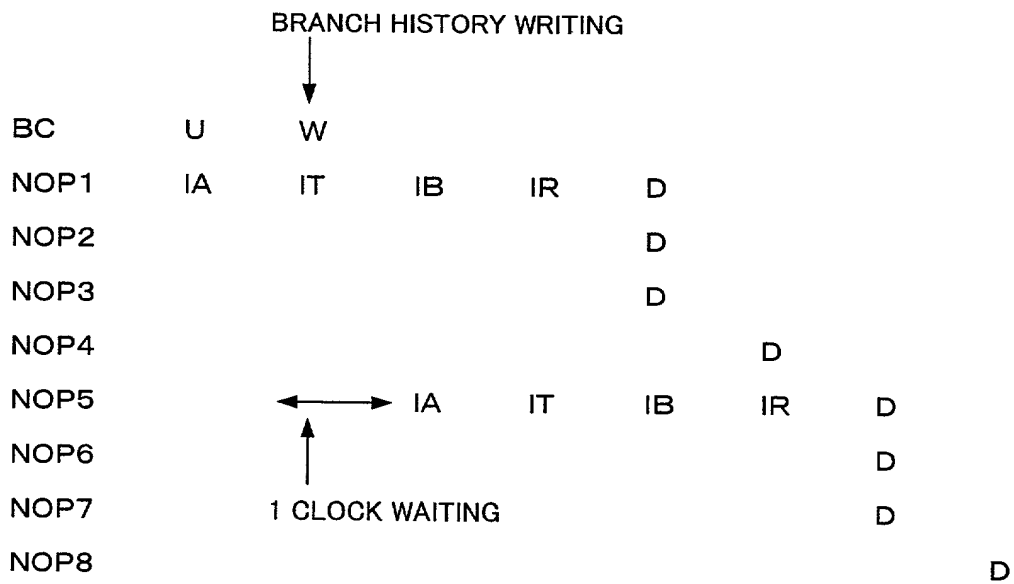


FIG. 1 Prior Art

001220" 5/22/66

# EXAMPLE OF SHORT LOOP

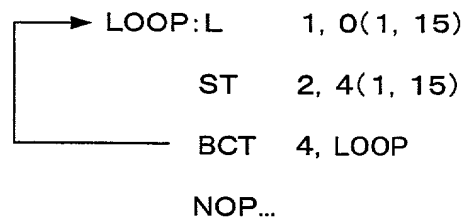


FIG. 2 A Prior Art

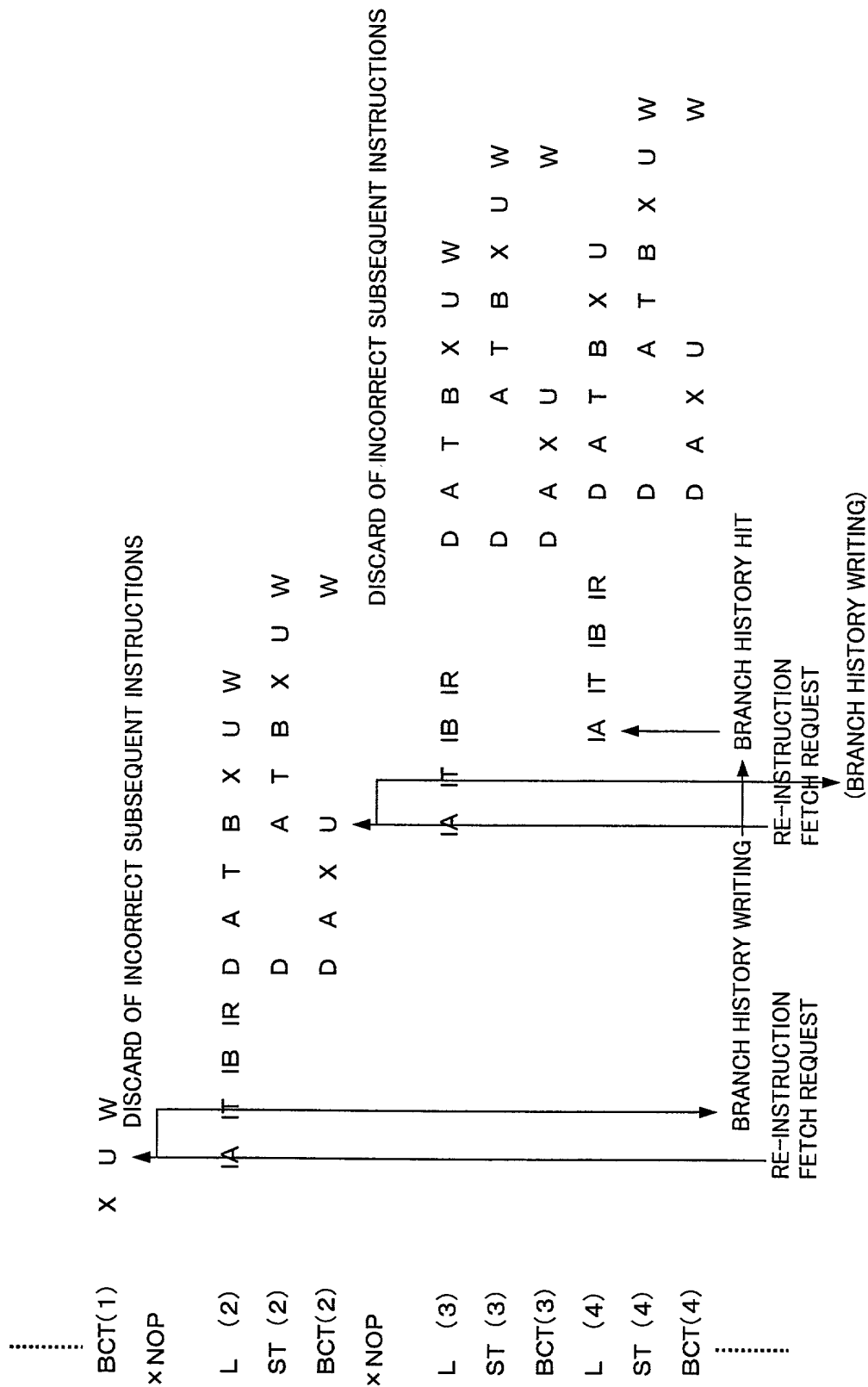
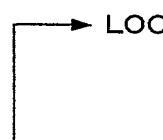


FIG. 2B Prior Art

BC	U	W					
NOP1	IA	IT	IB	IR	D		
NOP2					D		
NOP3					D		
NOP4						D	
NOP5		IA	IT	IB	IR	D	
NOP6						D	
NOP7							D
NOP8							D

**F I G. 3**

EXAMPLE OF SHORT LOOP



```
LOOP:L    1, 0(1, 15)
          ST    2, 4(1, 15)
          BCT   4, LOOP
          NOP...
```

FIG. 4A



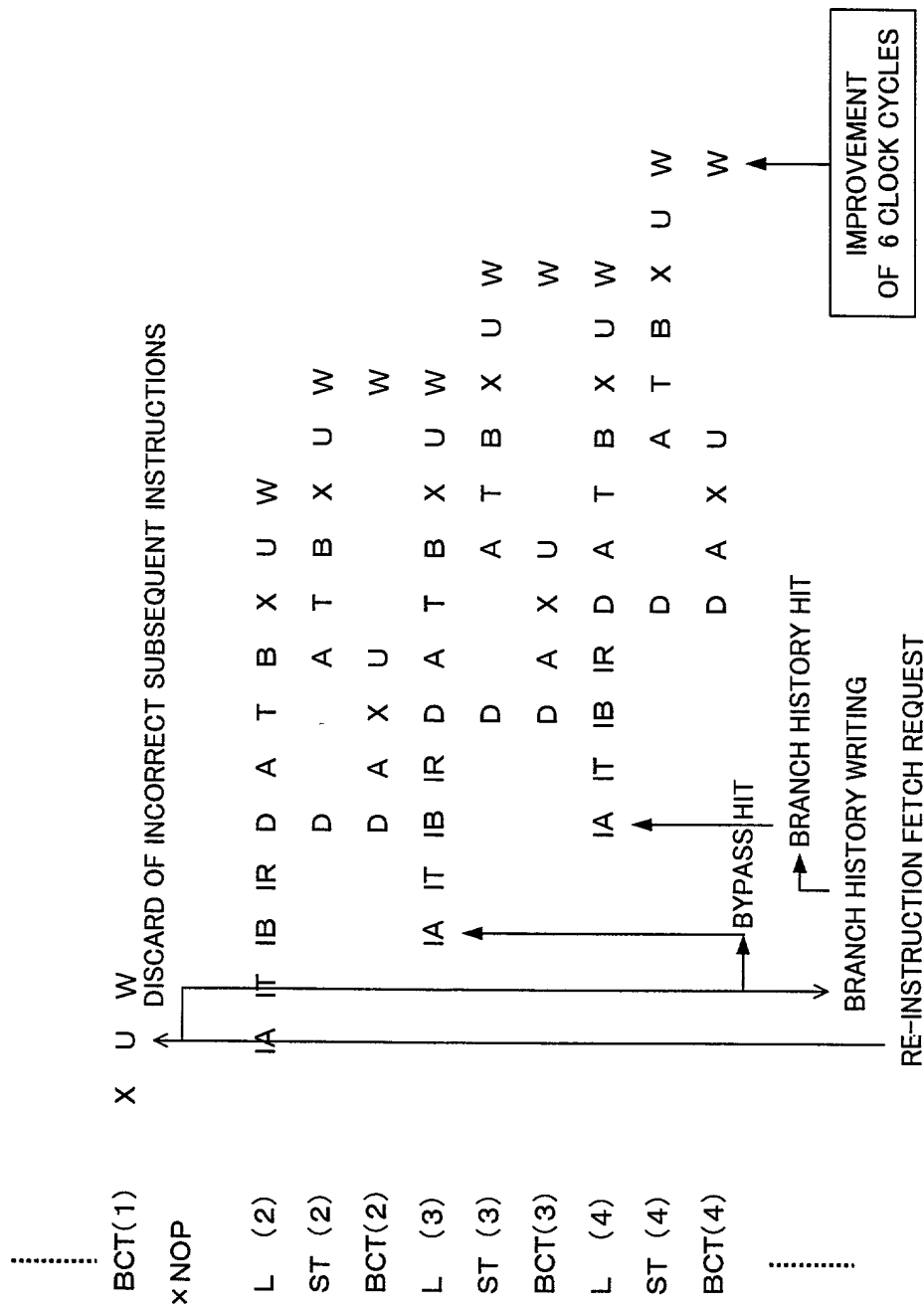


FIG. 4B

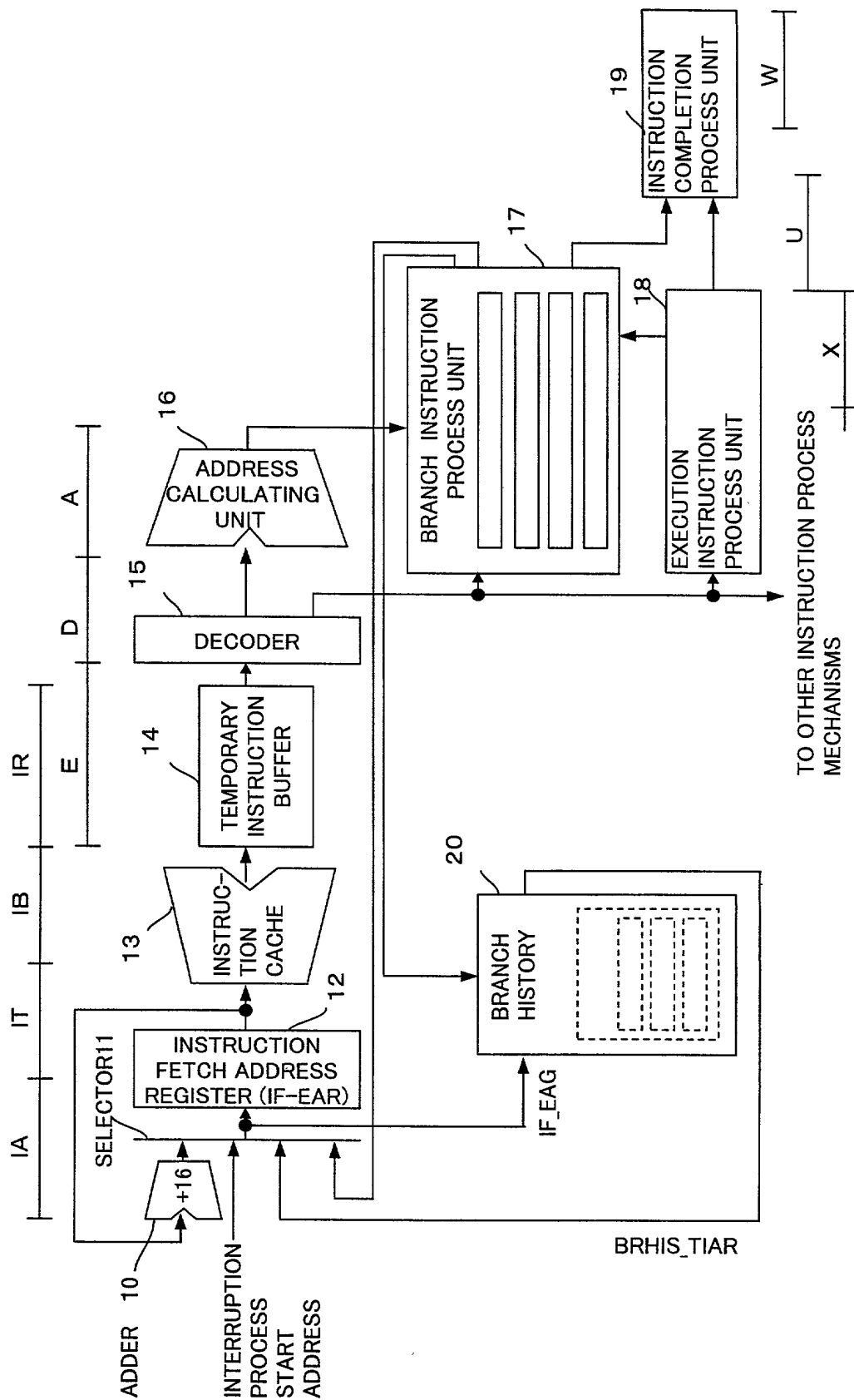


FIG. 5

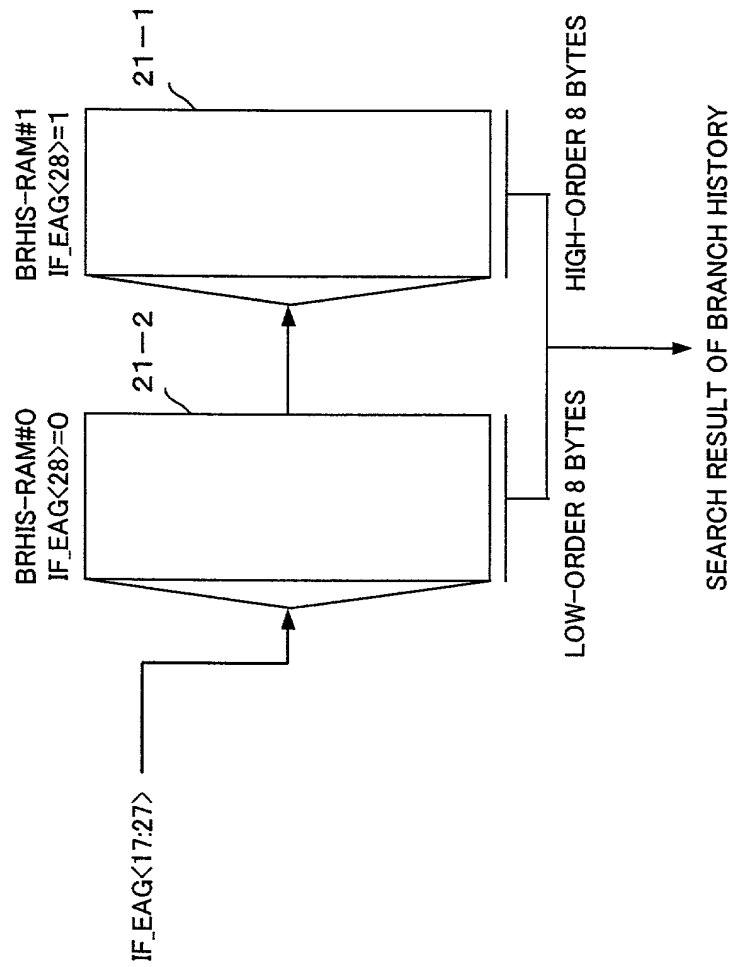


FIG. 6

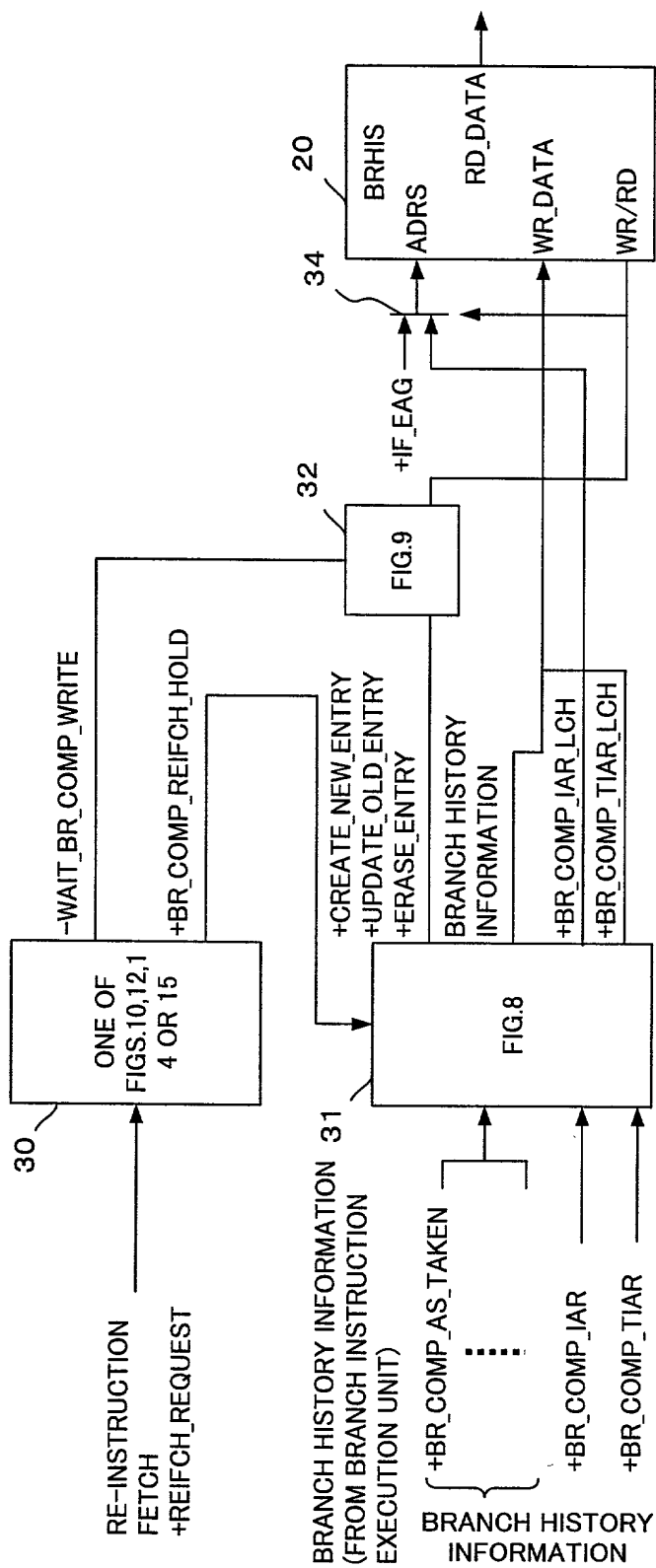


FIG. 7

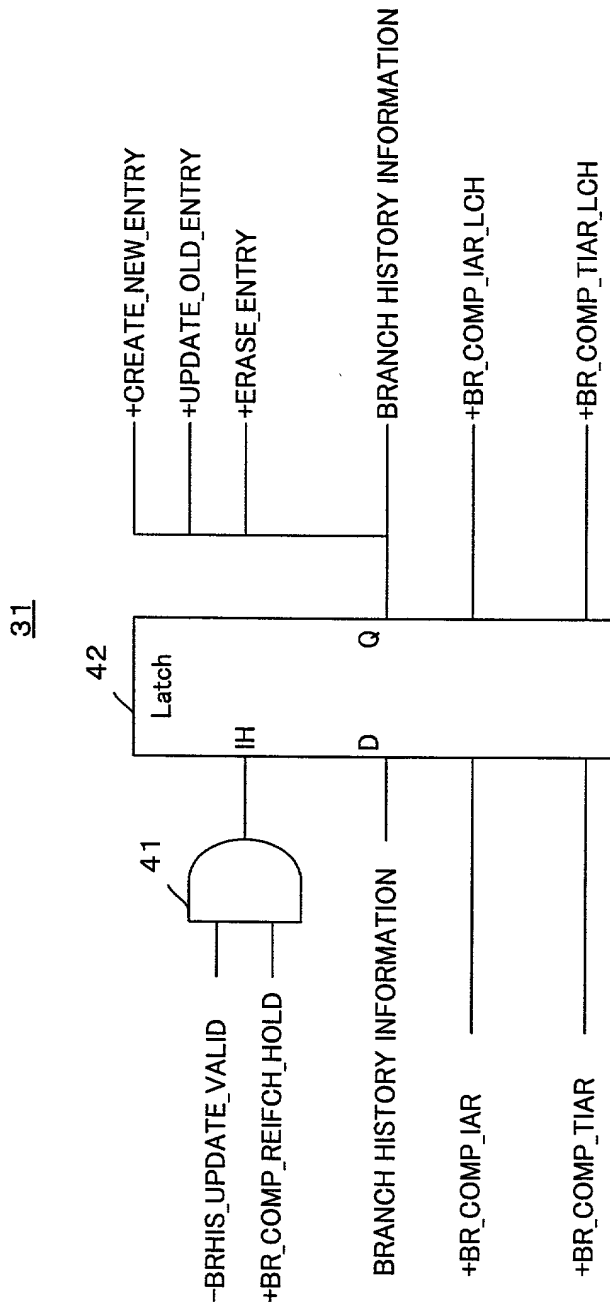
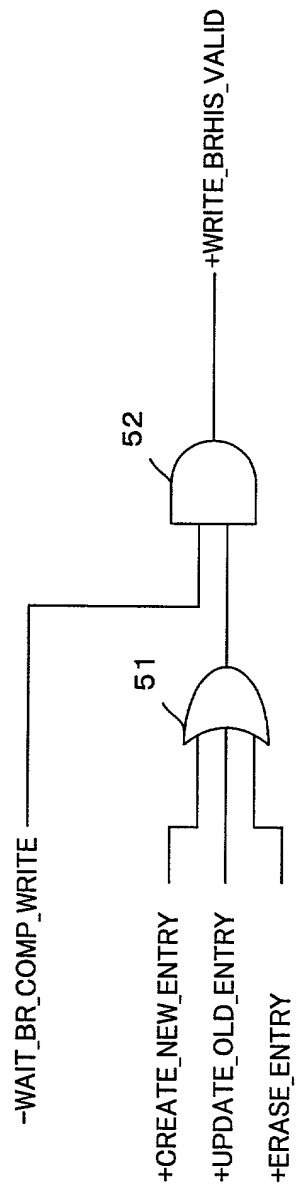


FIG. 8

32



F I G. 9

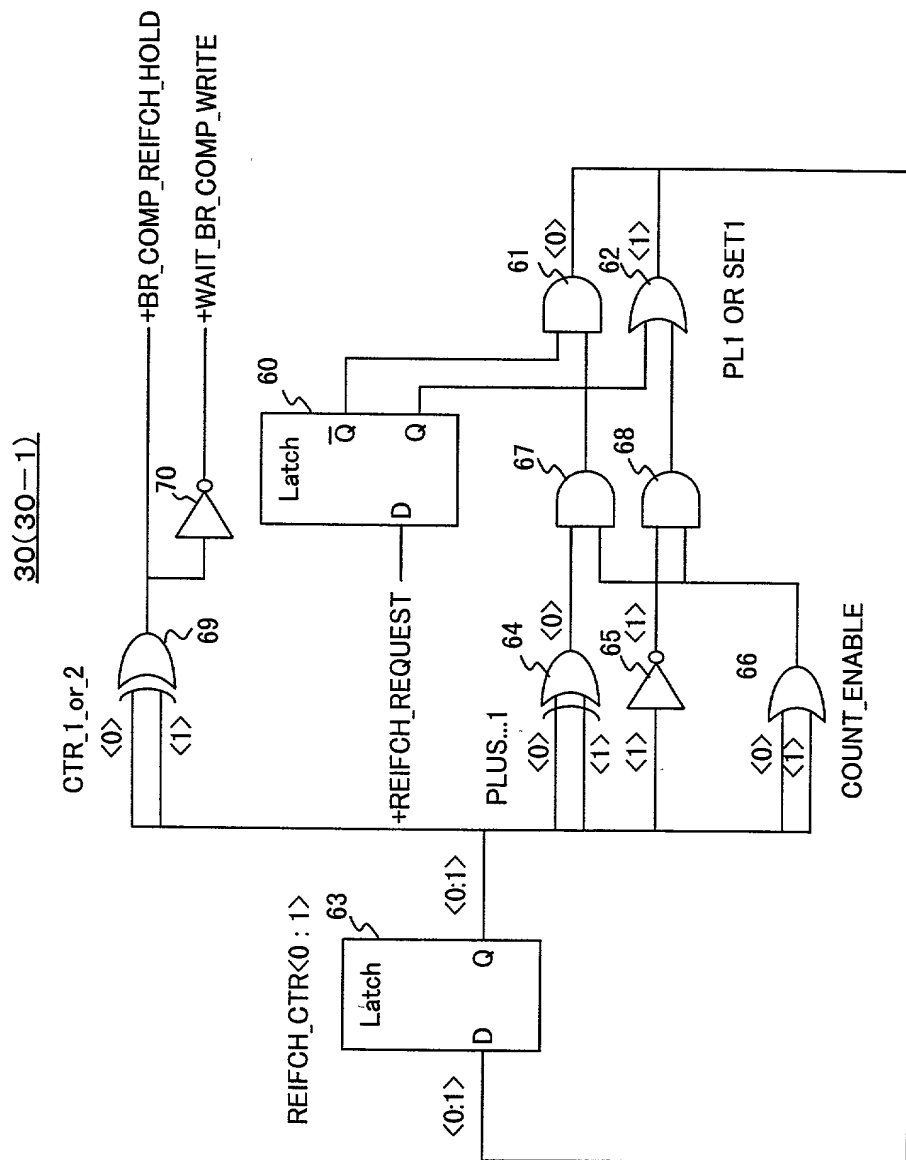


FIG. 10

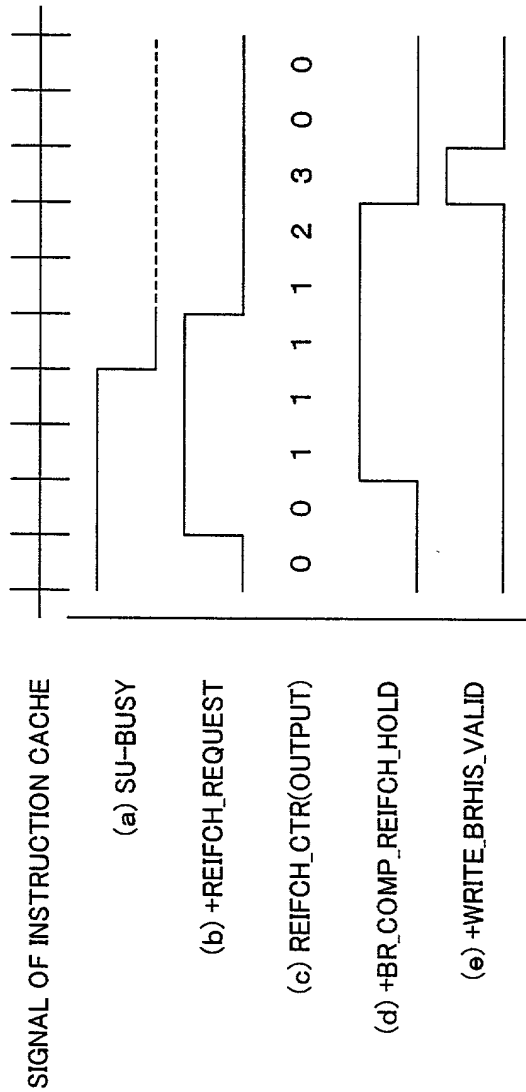


FIG. 11



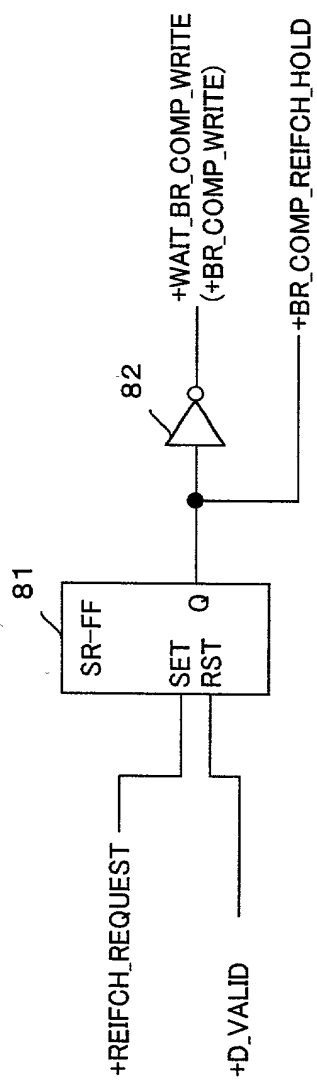


FIG. 12

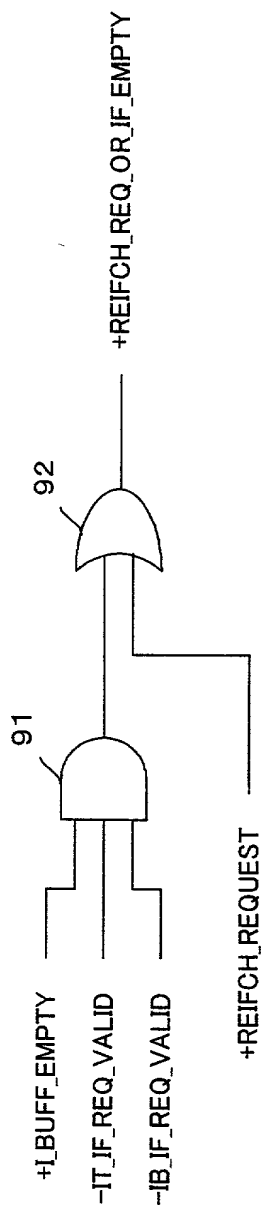


FIG. 13

30(30-3)

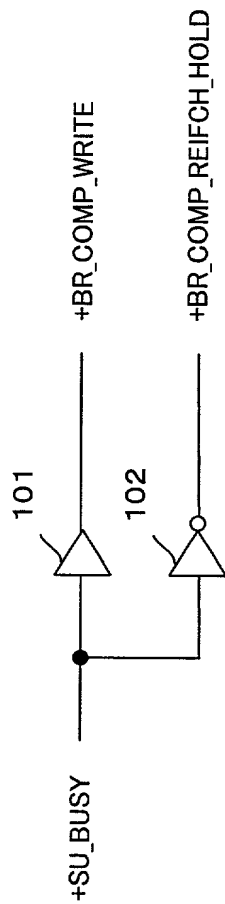


FIG. 14

30(30-4)

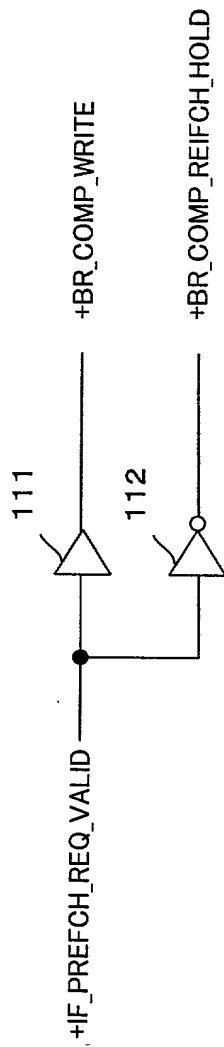


FIG. 15



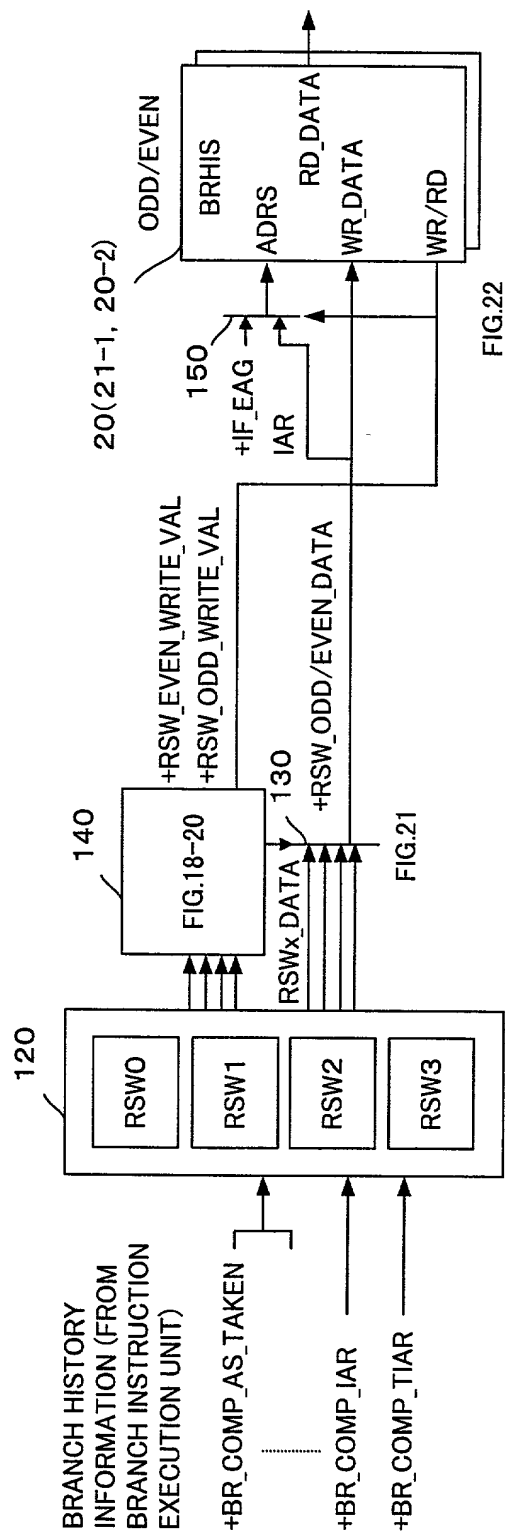


FIG. 17

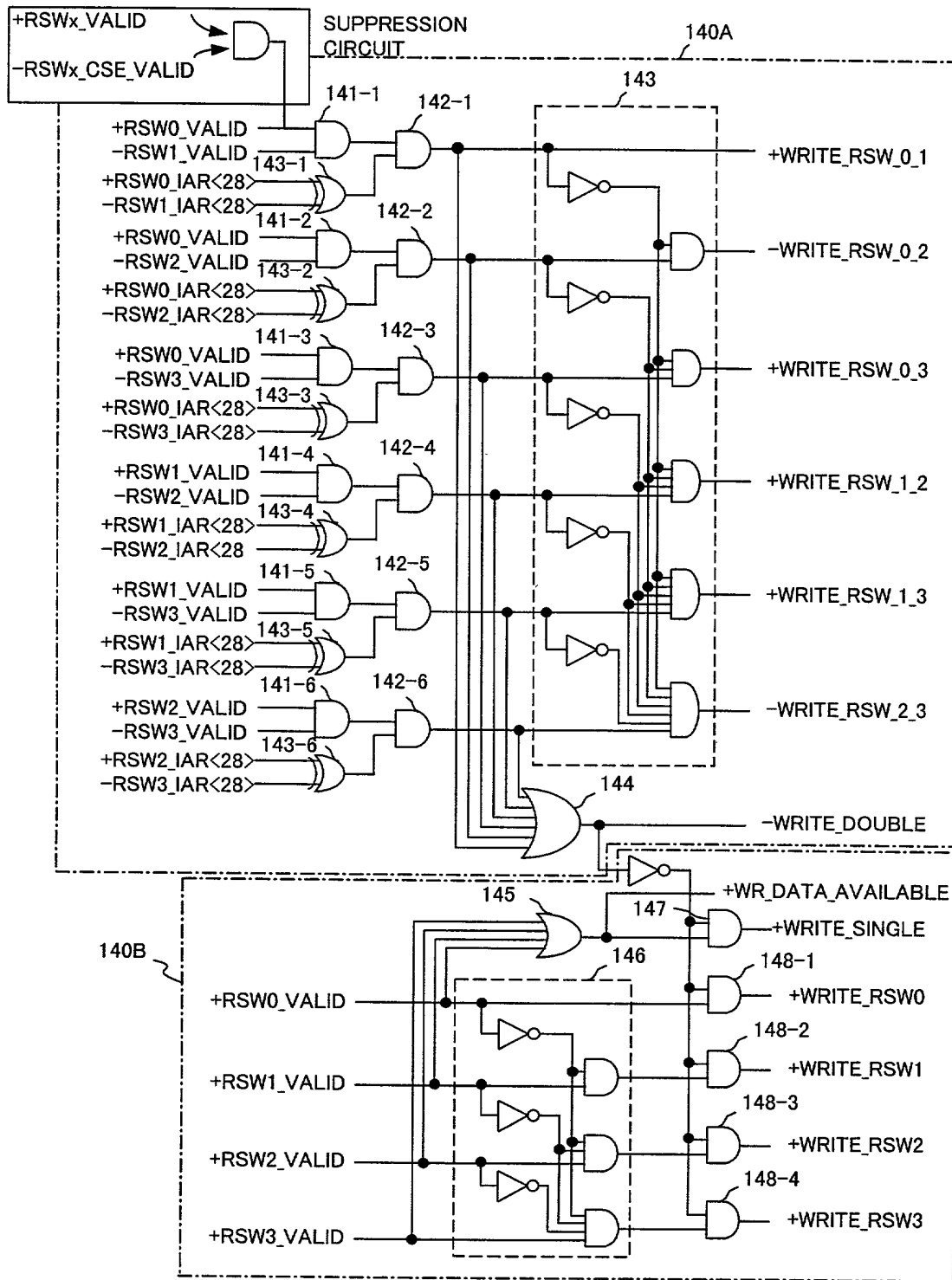


FIG. 18

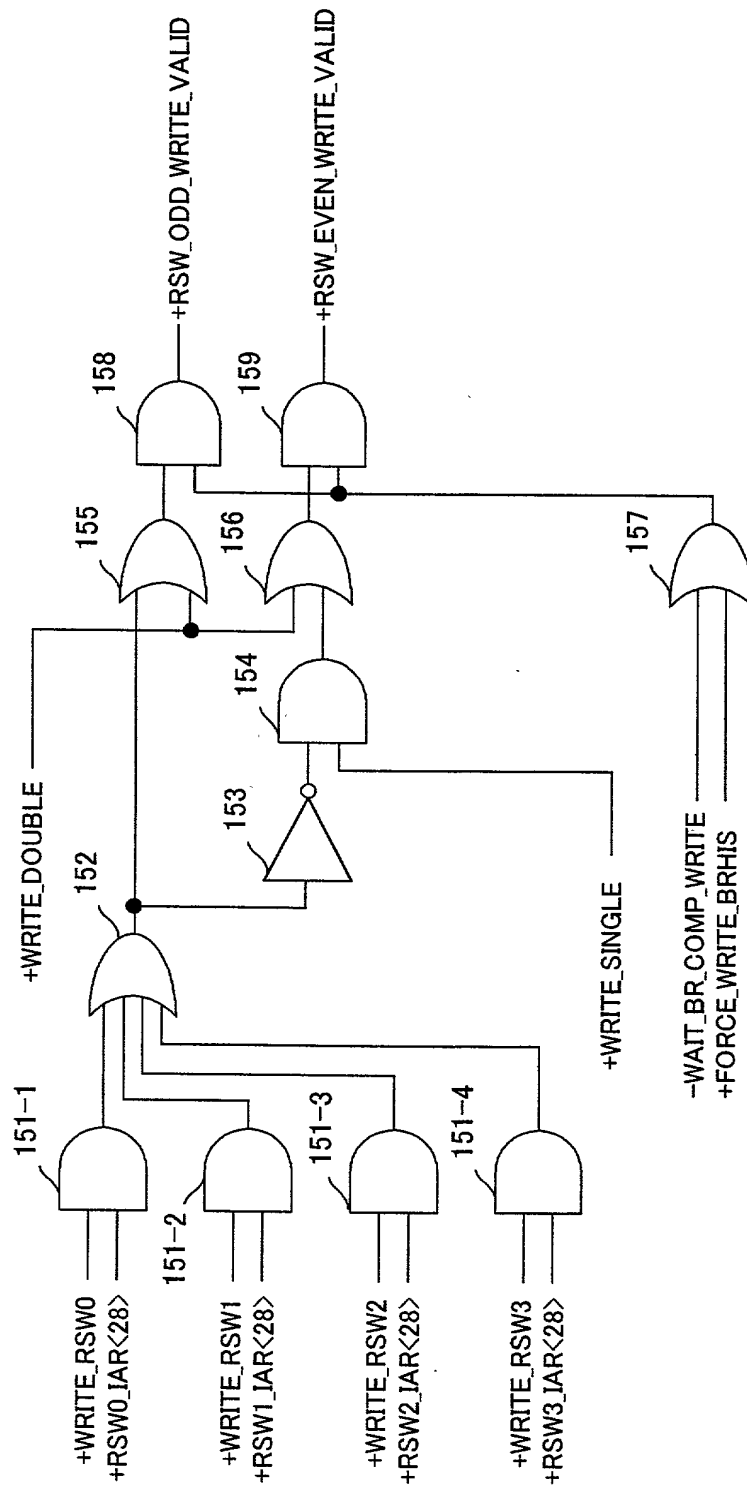


FIG. 19



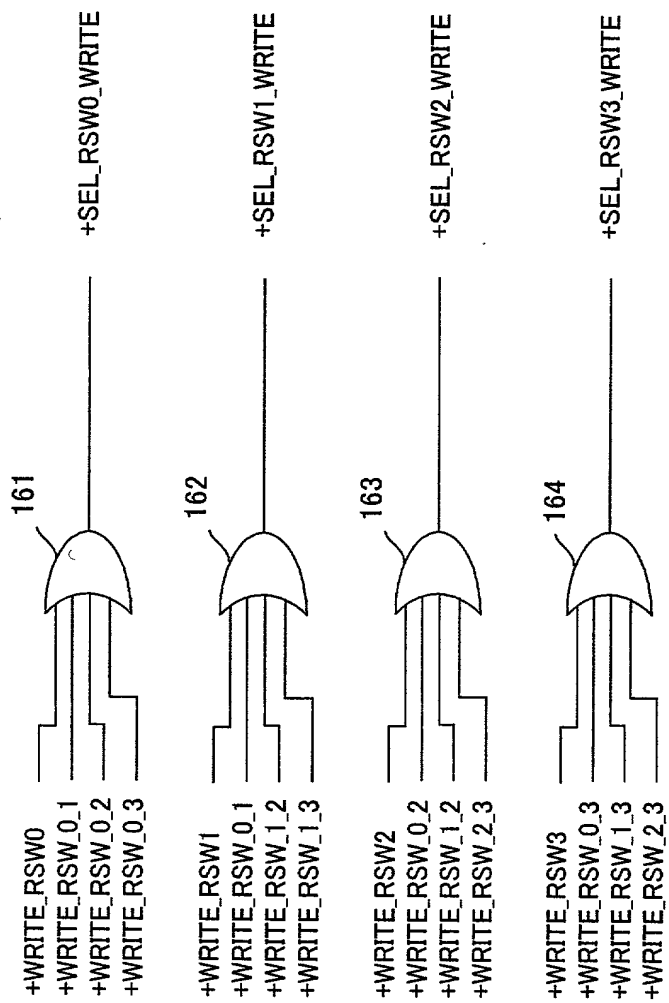


FIG. 20

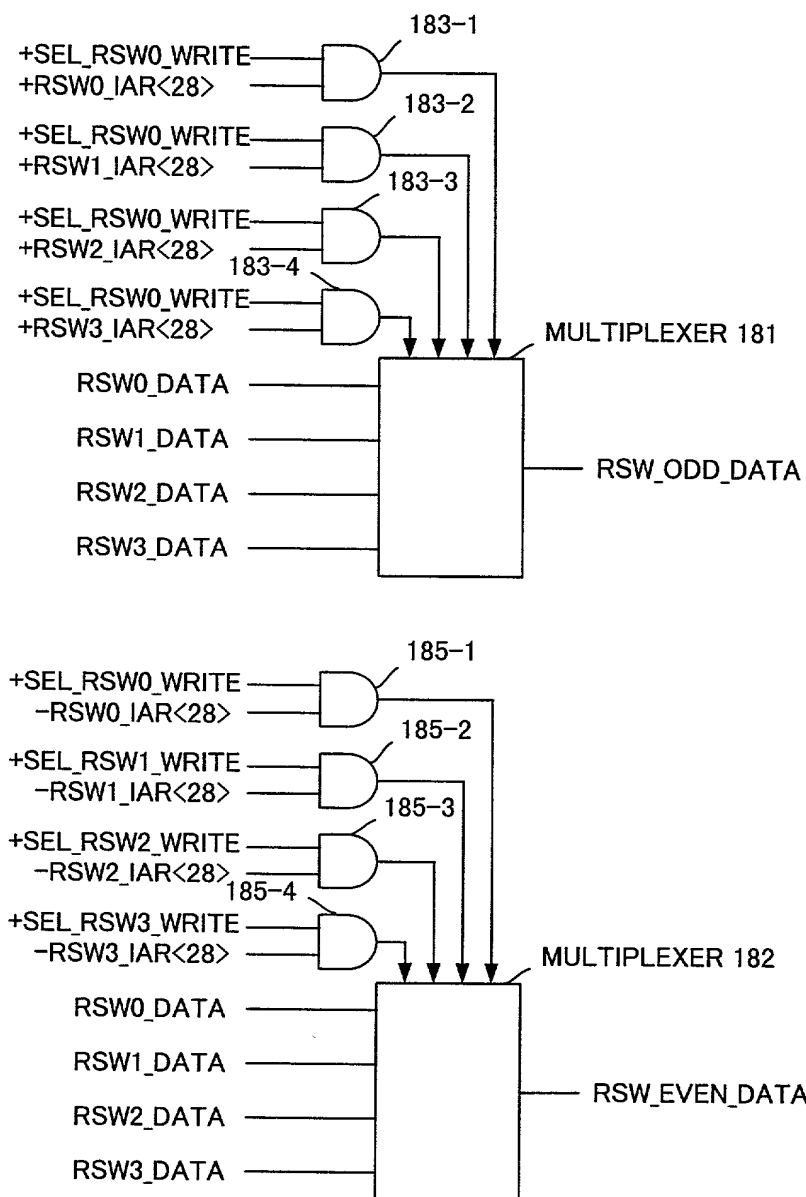


FIG. 21

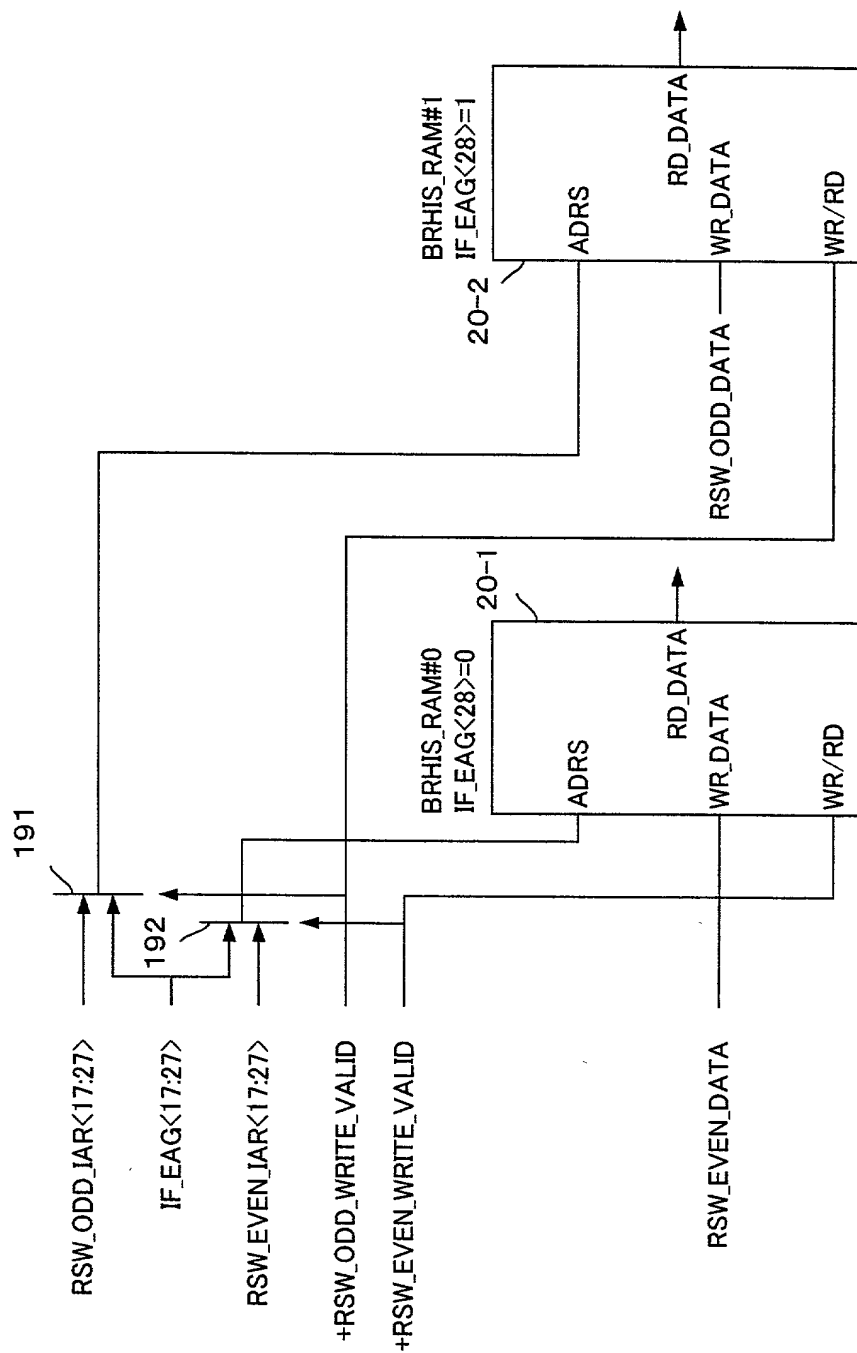


FIG. 22

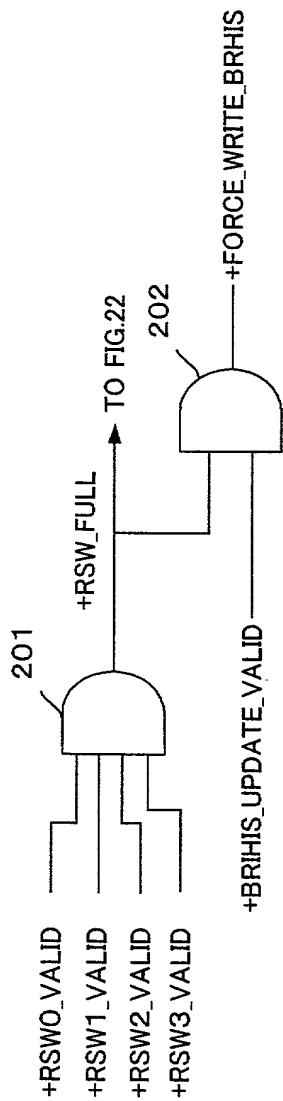


FIG. 23

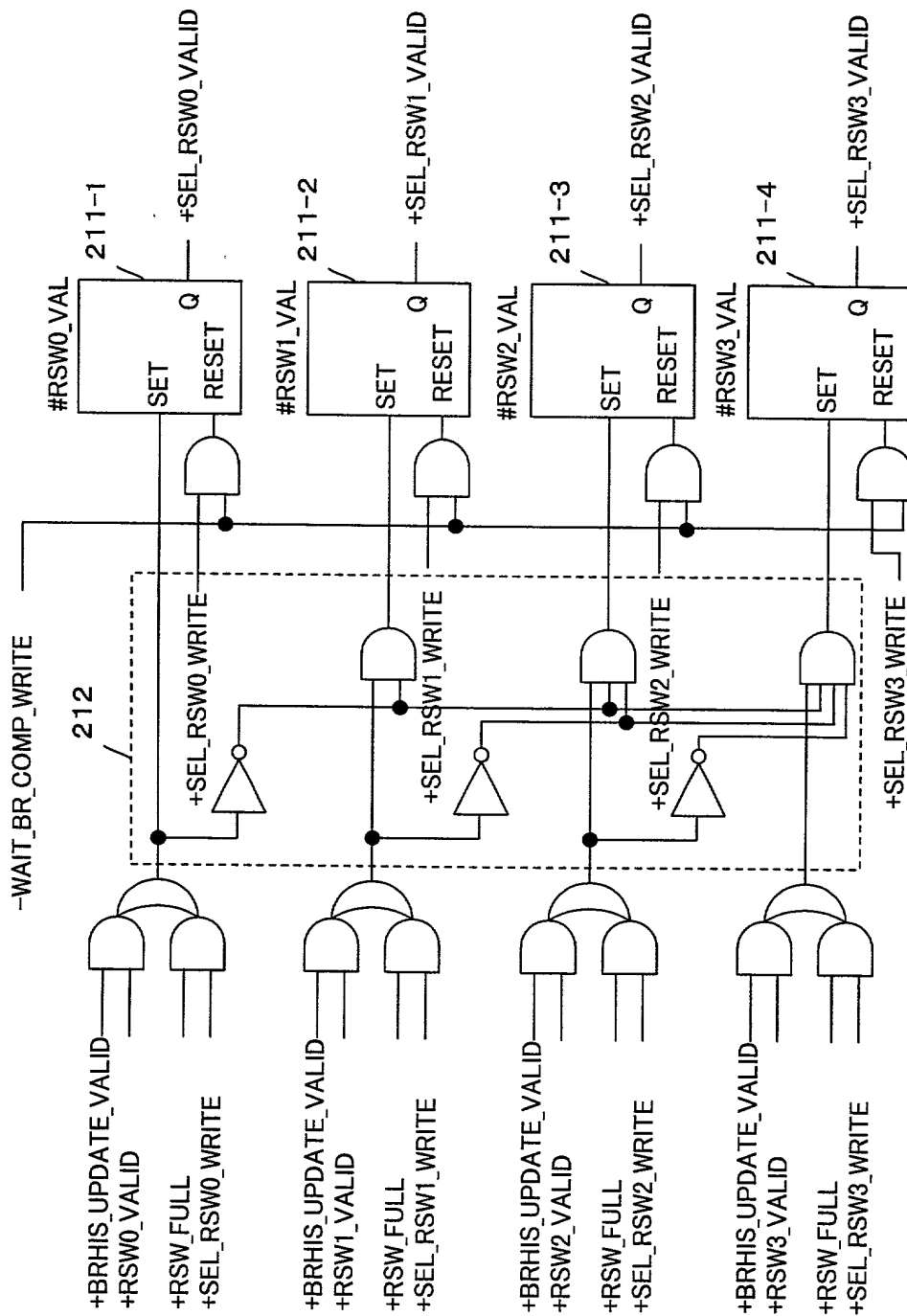


FIG. 24

120(120-2)

RSW 0	RSW 1	RSW 2	RSW 3
Valid	Valid	Valid	Valid
IAR	IAR	IAR	IAR
CSE-Valid	CSE-Valid	CSE-Valid	CSE-Valid
IID	IID	IID	IID
BRANCH HISTORY INFORMATION	BRANCH HISTORY INFORMATION	BRANCH HISTORY INFORMATION	BRANCH HISTORY INFORMATION

FIG. 25

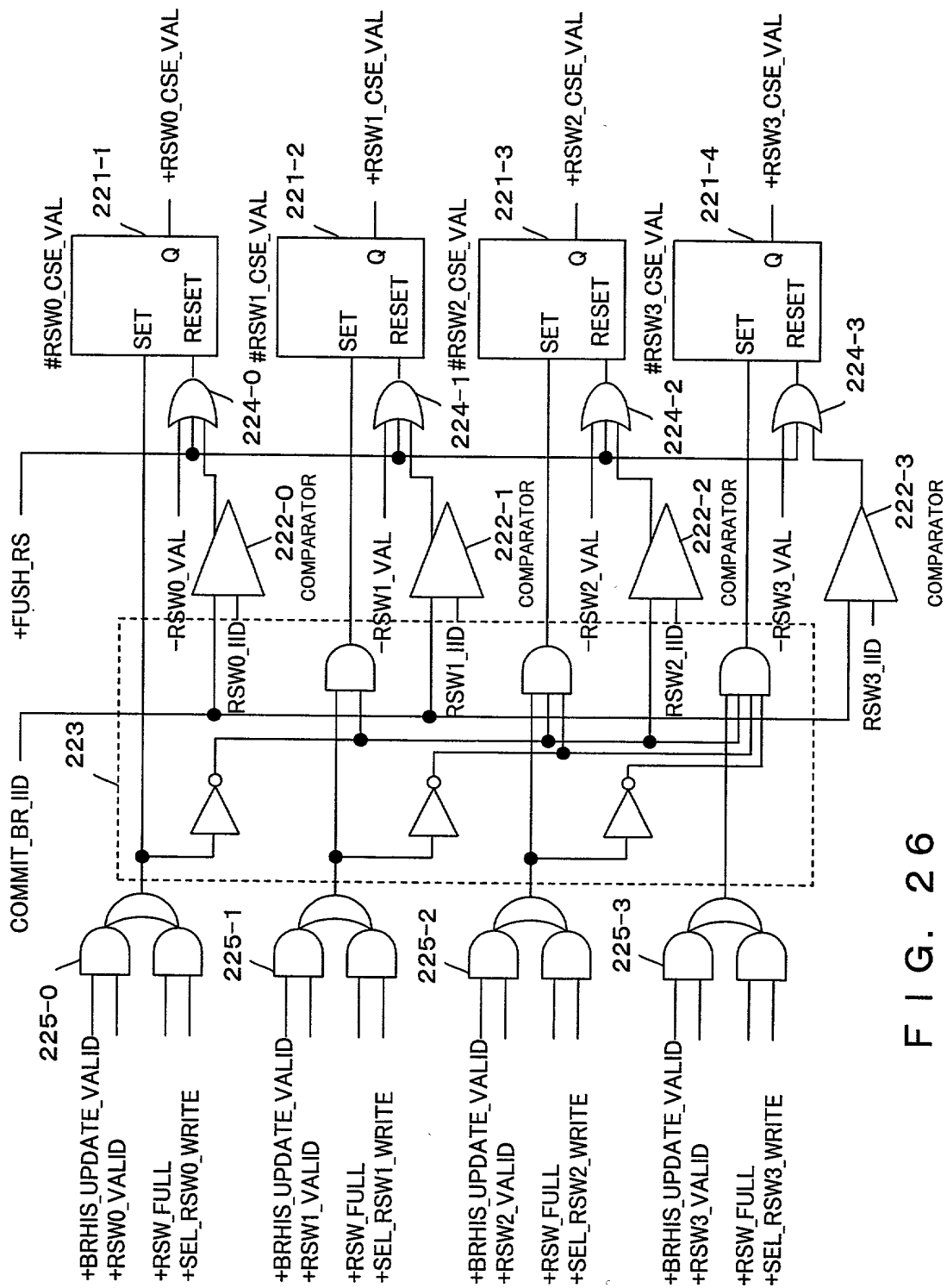


FIG. 26

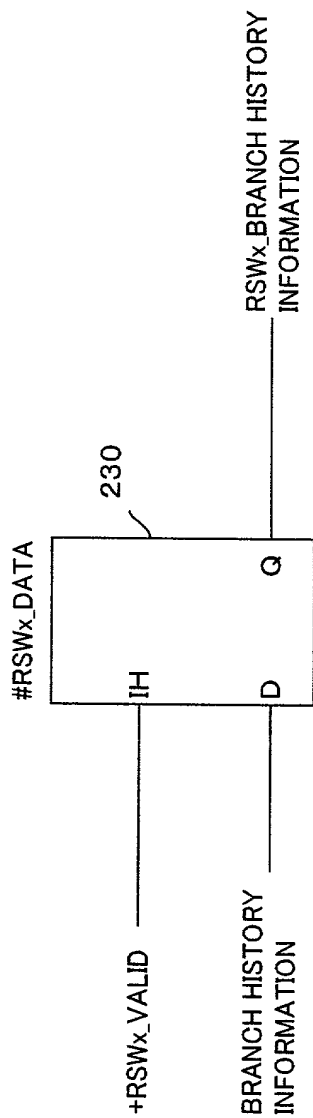


FIG. 27



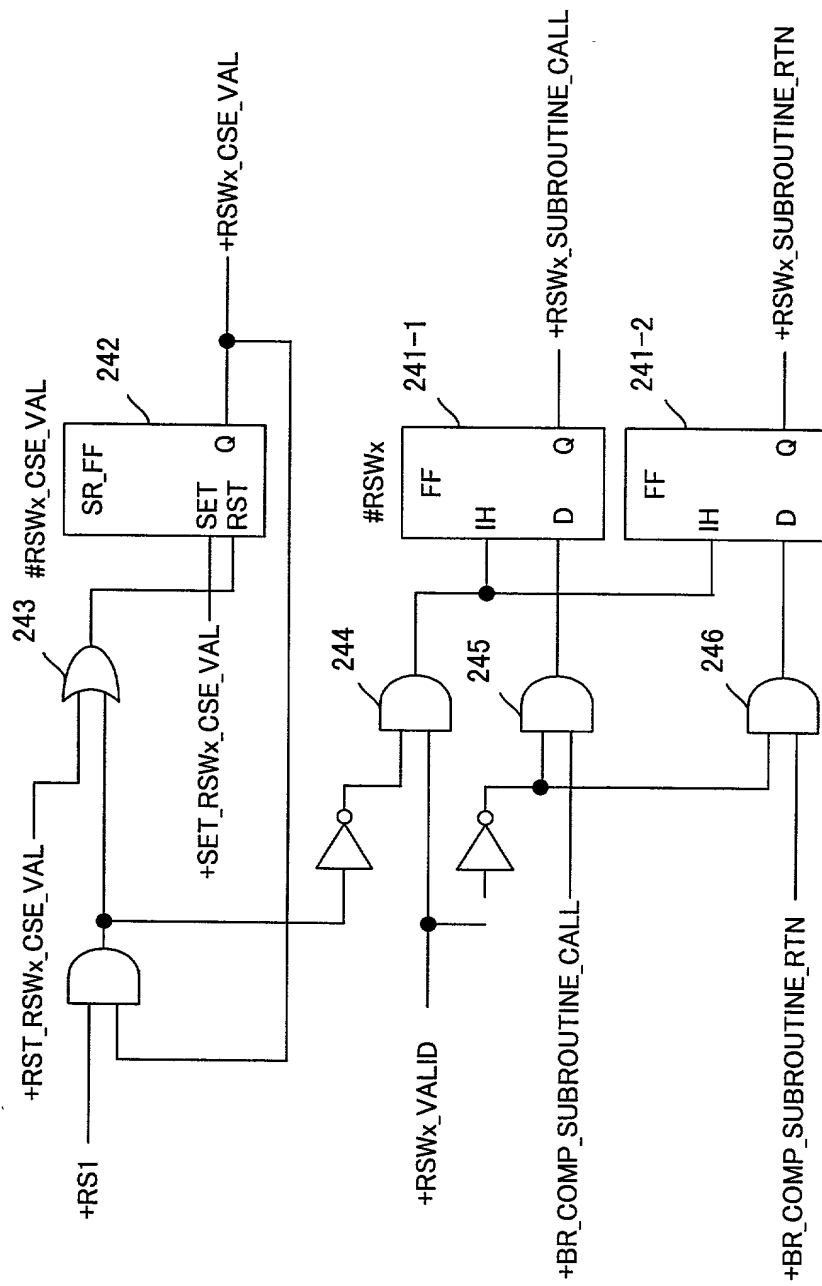


FIG. 28

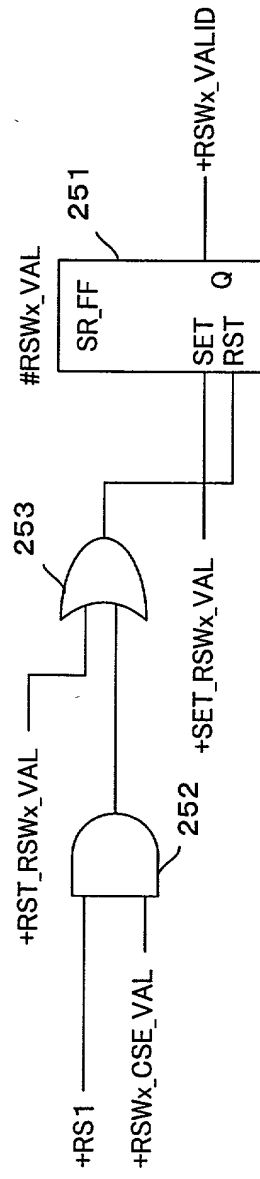


FIG. 29

001220" 9/22/85

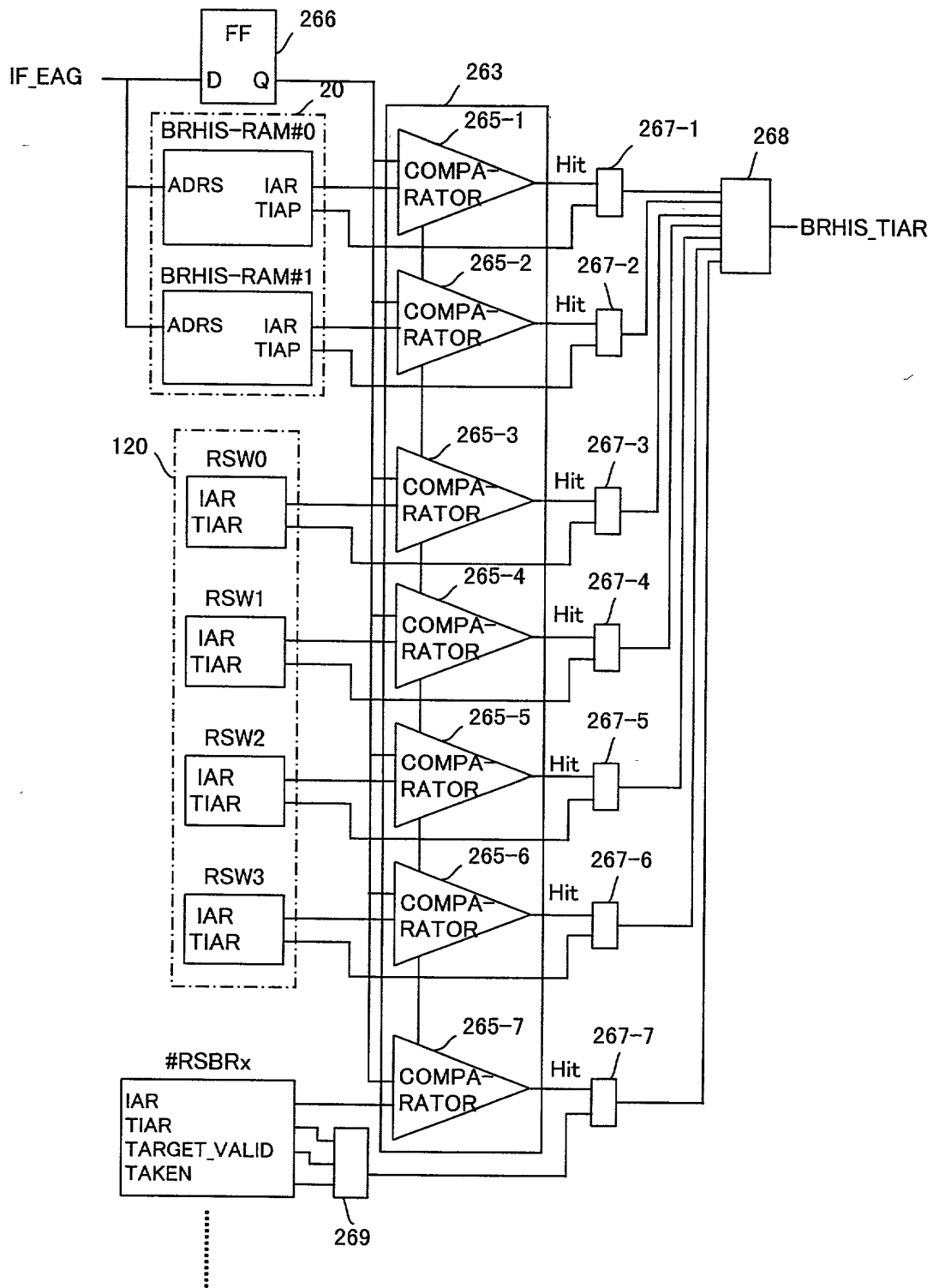


FIG. 30

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

## Declaration and Power of Attorney For Patent Application

特許出願宣言書及び委任状

Japanese Language Declaration

日本語宣言書

下記の氏名の発明者として、私は以下の通り宣言します。

As a below named inventor, I hereby declare that:

私の住所、私書箱、国籍は下記の私の氏名の後に記載された通りです。

My residence, post office address and citizenship are as stated next to my name.

下記の名称の発明に関して請求範囲に記載され、特許出願している発明内容について、私が最初かつ唯一の発明者（下記の氏名が一つの場合）もしくは最初かつ共同発明者であると（下記の名称が複数の場合）信じています。

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

DEVICE AND METHOD FOR CONTROLLING

WRITING OF BRANCH HISTORY INFORMATION

上記発明の明細書（下記の欄でxが1がついていない場合は、本書に添付）は、

the specification of which is attached hereto unless the following box is checked:

☐ 月 日に提出され、米国出願番号または特許協定条約国際出願番号を \_\_\_\_\_ とし、  
（該当する場合） \_\_\_\_\_ に訂正されました。

☐ was filed on \_\_\_\_\_  
as United States Application Number or  
PCT International Application Number  
\_\_\_\_\_ and was amended on  
\_\_\_\_\_ (if applicable).

私は、特許請求範囲を含む上記訂正後の明細書を検討し、内容を理解していることをここに表明します。

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

私は、連邦規則法典第37編第1条56項に定義されるとおり、特許資格の有無について重要な情報を開示する義務があることを認めます。

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

### Japanese Language Declaration (日本語宣言書)

私は、米国法典第35編119条(a)-(d)項又は365条(b)項に基づき下記の、米国外の国の少なくとも一ヶ国を指定している特許協力条約365(a)項に基づく国際出願、又は外国での特許出願もしくは発明者証の出願についての外国優先権をここに主張するとともに、優先権を主張している、本出願の前に出願された特許または発明者証の外国出願を以下に、枠内をマークすることで、示しています。

#### Prior Foreign Application(s)

外国での先行出願

11-277816

(Number)  
(番号)

Japan

(Country)  
(国名)

(Number)  
(番号)

(Country)  
(国名)

私、第35編米国法典119条(e)項に基づいて下記の米国外特許出願規定に記載された権利をここに主張いたします。

(Application No.)  
(出願番号)

(Filing Date)  
(出願日)

私は、下記の米国法典第35編120条に基づいて下記の米国外特許出願に記載された権利、又は米国外を指定している特許協力条約365条(c)に基づき権利をここに主張します。また、本出願の各請求範囲の内容が米国法典第35編112条第1項又は特許協力条約で規定された方法で先行する米国外特許出願に開示されていない限り、その先行米国外出願書提出日以降で本出願書の日本国内または特許協力条約国際提出日までの期間中に入手された、連邦規則法典第37編1条56項で定義された特許資格の有無に関する重要な情報について開示義務があることを認識しています。

(Application No.)  
(出願番号)

(Filing Date)  
(出願日)

(Application No.)  
(出願番号)

(Filing Date)  
(出願日)

私は、私自身の知識に基づいて本宣言書中で私が行なう表明が真実であり、かつ私の入手した情報と私の信じていること、さらに故意になされた虚偽の表明及びそれと同等の行為は米国法典第18編第1001条に基づき、罰金または拘禁、もしくはその両方により処罰されること、そしてそのような故意による虚偽の表明を行なえば、出願した、又は既に許可された特許の有効性が失われることを認識し、よってここに上記のごとく宣誓を致します。

I hereby claim foreign priority under Title 35, United States Code, Section 119 (a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT international application having a filing date before that of the application on which priority is claimed.

Priority Not Claimed

優先権主張なし

30th/September/1999

(Day/Month/Year Filed)  
(出願年月日)

(Day/Month/Year Filed)  
(出願年月日)

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below.

(Application No.)  
(出願番号)

(Filing Date)  
(出願日)

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s), or 365(c) of any PCT international application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.54 which became available between the filing date of the prior application and the national or PCT international filing date of application.

(Status: Patented, Pending, Abandoned)  
(現況: 特許許可済、係属中、放棄済)

(Status: Patented, Pending, Abandoned)  
(現況: 特許許可済、係属中、放棄済)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMS control number.

# Japanese Language Declaration (日本語宣言書)

委任状: 私は下記の発明者として、本出願に関する一切の  
 手続を米特許商標局に対して遂行する弁理士または代理人  
 として、下記の者を指名いたします。(弁理士、または代理  
 人の氏名及び登録番号を明記のこと)

POWER OF ATTORNEY: As a named inventor, I hereby appoint  
 the following attorney(s) and/or agent(s) to prosecute this  
 application and transact all business in the Patent and Trademark  
 Office connected therewith (list name and registration number)

James D. Halsey, Jr., 22,729; Harry John Staas, 22,010; David M. Pitcher, 25,908; John C. Garvey, 28,607; J. Randall Beckers,  
 30,358; William F. Herbert, 31,024; Richard A. Gollhofer, 31,106; Mark J. Henry, 36,162; Gene M. Garner II, 34,172; Michael D.  
 Stein, 37,240; Paul I. Kravetz, 35,230; Gerald P. Joyce, III, 37,648; Todd E. Marlene, 35,269; Harlan B. Williams, Jr., 34,756;  
 George N. Stevens, 36,938; Michael C. Soldner, 41,455; Norman L. Ourada, 41,235; Kevin R. Spivak, P-43,148; and William M.  
 Schertler, 35,348 (agent)

書類送付先

Send Correspondence to:

STAAS & HALSEY  
 700 Eleventh Street, N.W.  
 Suite 500  
 Washington, D.C. 20001

直接電話連絡先: (名前及び電話番号)

Direct Telephone Calls to: (name and telephone number)

STAAS & HALSEY  
 (202) 434-1500

唯一または第一発明者名	Full name of sole or first inventor
Masaki UKAI	
発明者の署名	Inventor's signature
日付	Date
Masaki Ukai	March 1, 2000
住所	Residence
Kanagawa, Japan	
国籍	Citizenship
Japan	
私書箱	Post Office Address
c/o FUJITSU LIMITED, 1-1, Kamikodanaka	
4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8588, Japan	
第二共同発明者	Full name of second joint inventor, if any
第二共同発明者	Second inventor's signature
日付	Date
住所	Residence
国籍	Citizenship
私書箱	Post Office Address

(第三以降の共同発明者についても同様に記載し、署名をす  
 ること)

(Supply similar information and signature for third and subsequent  
 joint inventors.)